October 1989

LIDS-TH-1922





**Research Supported By:**

*Joint Directors of Laboratories*
*Contract ONR/NOOO14-85-K-0782*

*Office of Naval Research*
*Contract ONR/NOOO14-84-K-0519*

# A CLASS OF DYNAMIC NONLINEAR RESOURCE ALLOCATION PROBLEMS

Patrick Ahamad Hosein



**Laboratory for Information and Decision Systems**
**MASSACHUSETTS INSTITUTE OF TECHNOLOGY, CAMBRIDGE, MASSACHUSETTS 02139**

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S)<br>LIDS-TH-1922 | 5. MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a. NAME OF PERFORMING ORGANIZATION<br>Lab. for Inf. and Dec. Systems<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>Office of Naval Research |
| 6c. ADDRESS (City, State, and ZIP Code)<br>Mass. Institute of Technology<br>Room 35-406, LIDS<br>Cambridge, MA 02139 | 7b. ADDRESS (City, State, and ZIP Code)<br>800 N. Quincy Street<br>Arlington, VA 22217-5000 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>N00014-85-K- 0782 |
| 8c. ADDRESS (City, State, and ZIP Code) | 10. SOURCE OF FUNDING NUMBERS<br>PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. NR564-001 / WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification)
A CLASS OF DYNAMIC NONLINEAR RESOURCE ALLOCATION PROBLEMS

12. PERSONAL AUTHOR(S) PATRICK A. HOSEIN

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| Thesis Ph.D. | FROM ______ TO ______ | Oct. 1989 | 214 |

16. SUPPLEMENTARY NOTATION

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

We consider a class of dynamic resource allocation problems a specific example of which is the Weapon-Target Assignment problem. This problem is concerned with the optimal assignment of resources in a military engagement. These problems are, in general, NP-Complete, so our aim is to provide insight into the problem and its solution. We will provide analytical results for simple cases of the problem. We will also provide sub-optimal algorithms, together with bounds on the objective function, under certain assumptions.

The battle scenario of the military engagement being modeled is as follows. The offense launches a number of weapons (the targets) which are aimed at valuable assets of the defense. The defense has a number of defensive weapons each of which can engage at most one target. The outcome of such an engagement is stochastic. In the static scenario all weapons are fired simultaneously. In the dynamic scenario some weapons are assigned and fired and the outcomes of these engagements are observed before further assignments are made. Two objective functions will be considered. In the Target Based problem, corresponding to weighted subtractive defens

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION<br>unclassified (15P) |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL<br>CDR T. WELLS | 22b. TELEPHONE (Include Area Code) (202)696-4713 / 22c. OFFICE SYMBOL Code 1224 |

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

U.S. Government Printing Office: 1985-507-047

ABSTRACT

values are assigned to the targets and the objective is to assign weapons to targets so as to minimize the total expected value of the surviving targets after all weapons have been fired. In the Asset-Based problem, corresponding to preferential defense, we assume that each target is aimed to an asset and, if not intercepted, destroys it with some given probability. Values are assigned to each asset and the objective is to assign weapons to targets so as to maximize the total expected value of the surviving assets.

We will present analytical results for simple cases of the Dynamic Target-Based problem as well as asymptotic results as the number of targets goes to infinity. A sub-optimal algorithm for the Static Asset-Based problem together with an analytical bound on the optimal value will also be provided. A sub-optimal algorithm for the Dynamic Asset-Based problem will be presented together with a computational bound on the optimal value. Several numerical and sensitivity analysis results will be given. Generally, under suitable assumptions, we show that dynamic strategies can approximately double the defense effectiveness as compared to their static counterparts.

OCTOBER 1989 LIDS-TH-1922

# A CLASS OF DYNAMIC NONLINEAR RESOURCE ALLOCATION PROBLEMS

By

Patrick Ahamad Hosein

This report is based on the unaltered thesis of Patrick A. Hosein submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy at the Massachusetts Institute of Technology in October 1989. The research was conducted at the M.I.T. Laboratory for Information and Decision Systems with partial support provided by the Joint Directors of Laboratories under contract ONR/N00014-85-K-0782 and by the Office of Naval Research under contract ONR/N00014-84-K-0519.



Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

# A CLASS OF DYNAMIC NONLINEAR RESOURCE ALLOCATION PROBLEMS

by

Patrick Ahamad Hosein

B.S., Massachusetts Institute of Technology, 1983

S.M., Massachusetts Institute of Technology, 1985

Submitted to the Department of

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September, 1989



Signature of Author: ______________________

Department of Electrical Engineering and Computer Science
September, 1989

Certified by: ______________________

Michael Athans
Thesis Supervisor

Accepted by: ______________________

Arthur C. Smith
Chairman, Departmental Graduate Committee

# A Class of Dynamic Nonlinear Resource Allocation Problems

by

Patrick Ahamad Hosein

Submitted to the Department of Electrical Engineering and Computer Science on September 29, 1989 in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

## ABSTRACT

We consider a class of dynamic resource allocation problems a specific example of which is the *Weapon-Target Assignment* problem. This problem is concerned with the optimal assignment of resources in a military engagement. These problems are, in general, NP-Complete, so our aim is to provide insight into the problem and its solution. We will provide analytical results for simple cases of the problem. We will also provide sub-optimal algorithms, together with bounds on the objective function, under certain assumptions.

The battle scenario of the military engagement being modeled is as follows. The offense launches a number of weapons (the targets) which are aimed at valuable assets of the defense. The defense has a number of defensive weapons each of which can engage at most one target. The outcome of such an engagement is stochastic. In the *static* scenario all weapons are fired simultaneously. In the *dynamic* scenario some weapons are assigned and fired and the outcomes of these engagements are observed before further assignments are made. Two objective functions will be considered. In the *Target-Based* problem, corresponding to weighted subtractive defense, values are assigned to the targets and the objective is to assign weapons to targets so as to minimize the total expected value of the surviving targets after all weapons have been fired. In the *Asset-Based* problem, corresponding to preferential defense, we assume that each target is aimed at an asset and, if not intercepted, destroys it with some given probability. Values are assigned to each asset and the objective is to assign weapons to targets so as to maximize the total expected value of the surviving assets.

We will present analytical results for simple cases of the Dynamic Target-Based problem as well as asymptotic results as the number of targets goes to infinity. A sub-optimal algorithm for the Static Asset-Based problem together with an analytical bound on the optimal value will also be provided. A sub-optimal algorithm for the Dynamic Asset-Based problem will be presented together with a computational bound on the optimal value. Several numerical and sensitivity analysis results will be given. Generally, under suitable assumptions, we show that dynamic strategies can approximately double the defense effectiveness as compared to their static counterparts.

Thesis Supervisor: Dr. Michael Athans
Title: Professor of Systems Science and Engineering

# Acknowledgements

I would like to thank my thesis supervisor, Professor Michael Athans. His guidance and support played a major role in the development of the research presented in this thesis. He has also had a direct influence on my professional development.

I would like to thank the other members of my thesis committee, Professor Dimitri Bertsekas and Professor James Orlin, for their helpful comments and advice. I also wish to thank Dr. David Castañon for our discussions on my research and his helpful ideas. He also read my thesis and made several helpful suggestions. I would also like to thank my friend Jim Walton for our numerous discussions on our research.

Two special people helped me through the difficult times; my wife and my son. I would like to thank my wife for all of her sacrifices and I would like to thank my son for always being able to make me smile. My parents, sisters and brother also provided support and encouragement during my years at MIT.

Throughout my years at MIT I have had many friends who have made my stay enjoyable. These include, Brian Yokum, Malcolm Duke, David Lau, Anthony D'amico, Christine Shadle, John Spinelli, Ellen Hahne, Paul Tseng, Jeff Shamma, Tony Rodriguez, Jason Papastavrou and Madan Venugopal. I would like to thank them all as well as all of my friends at the Laboratory for Information and Decision Systems. I also wish to thank some additional MIT professors who have had a direct influence on my education, Professor Amar Bose, Professor Ken Stevens and Professor David Staelin. Finally, I wish to thank the staff at LIDS, especially Fifa Monserrate, for their many favors.

This research was conducted at the MIT Laboratory for Information and Decision Systems with partial support provided by the Joint Directors of Laboratories under contract ONR/N00014-85-K-0782 and by the Office of Naval Research under contract ONR/N00014-84-K-0519.

*TO NICHOLAS*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

A resource allocation problem is one in which a set of resources must be allocated to a set of utilities so as to optimize some given criterion. Some examples of resource allocation problems include the allocation of personnel to jobs (e.g. the assignment of nurses to shifts), the allocation of machines to tasks (e.g. in manufacturing) and the assignment of weapons to targets in a military conflict. These problems can be stated as mathematical optimization problems and solved. The weapon to target assignment problem, the main focus of this thesis, will be discussed in detail in the next section.

The class of problems we will consider has certain basic properties which restrict the solution methods that can be used. Resources can only be assigned in integral quantities. Problems with this property are called Integer Programming problems. Note that under this restriction the number of feasible solutions is finite. However, we will be concerned with Large-Scale problems. For these problems the number of feasible solutions is so large that complete enumeration and evaluation of each feasible solution is an impractical option.

Another property of the class of problems to be considered is that the resources will be assumed to be error prone. In other words, the action of a resource on a utility has a stochastic outcome. With some given probability the resource will have no effect on the utility. Because of this, the objective to be optimized will be the expected value of some performance measure.

We will also be looking at problems in which resources are assigned in time stages. In each stage some of the available resources will be assigned to utilities. The effect of these resources on the utilities will be observed before the assignments in the following stage are made. Note that these observations are useful because of the failure prone nature of the resources. If the resources were not failure prone then the outcomes would be deterministic and can be determined in advance. Resources will also be assumed to be non-renewable so that once one is used it cannot be assigned in a later stage.

Most of the resource allocation problems that have been studied in the literature have a linear objective function. In other words the benefit of each utility increases linearly with the number of resources assigned to it. The resource allocation models in this thesis all have nonlinear objectives. One of these objective functions will be convex. The other will be neither convex nor concave.

The issues which arise in solving the problems with these properties can best be discussed if we look at a specific class of problems. In the next section we will describe a motivating example for our research, the weapon to target assignment problem. Throughout the thesis we will use this example to illustrate our results.

## 1.1 Motivating Example

In this section we will describe the weapon to target allocation problem. This problem, which is used to model the defense of assets in a military conflict, can be described as follows. The offense (the enemy) launches a number of offensive weapons which are aimed at valuable assets of the defense. Since these weapons will be the targets of the defense's weapons, henceforth we will call them targets. Each of these targets is aimed at exactly one of the defense's assets and, if it is not intercepted, it destroys the asset with some lethality probability. We will assume that the impact of a target on an asset is independent of all other targets and assets. The defense has a number of defensive weapons with which to engage these incoming targets. The engagement of a a target by a weapon will be modeled as a stochastic event. A probability, called a kill probability, will be assigned to each weapon-target pair. This will be the probability that the weapon destroys the target if it is assigned to it. We will assume that the engagement of a weapon-target pair is independent of all other weapons and targets. Note that a particular target may be engaged by more than one weapon in a particular stage (Salvo attacks).[1]

Two different objectives of the defense will be considered. In the Target-Based version of the problem, values will be assigned to the incoming offensive weapons and the objective is to assign defensive weapons to these targets so as to minimize the expected total value of the targets which survive after all engagements. Target-Based problems correspond to what is called weighted subtractive defense. In the Asset-Based version of the problem, values are assigned to the defended assets and the objective of the defense is to assign weapons to targets so as to maximize the expected total value of the assets which survive the offense's attack. Asset-Based problems lead to preferential defense tactics. One can show that the Target-Based version of the problem is a special case of the Asset-Based problem in which exactly one target is aimed at each of the assets. If the value of the asset is assigned to the target aimed for it, then the corresponding Target-Based problem is equivalent to the Asset-Based one.

In the static versions of the Target-Based and Asset-Based problems we will assume that all weapons are assigned and fired simultaneously. Damage assessment is made after all weapon-target engagements. In the case of the Target-Based problem this is the assessment of the set of surviving targets while in the Asset-Based problem it is the assessment of the set of destroyed assets. We will refer to these as the *Static Target-Based Weapon-Target Allocation Problem* and the *Static Asset-Based Weapon-Target Allocation Problem.*

The major focus of this thesis is the analysis of the dynamic versions of the Target-Based and Asset-Based problems. In the dynamic problem, weapons are allocated in stages with the assumption that the outcomes (i.e. survival or destruction of each target) of the weapon-target engagements of the previous stage are observed (perfectly) before assignments for the present stage are made. We will assume that each weapon can be used only once.

[1]The weapon-target allocation problem is but one of the many problems that need to be addressed in the field of Command and Control ($C^2$) theory. The perspectives paper by Athans [1] presents some of the other basic problems in the theory of $C^2$ systems.

Note that, in practice, it is not possible to observe perfectly the outcome of each weapon-target engagement. This is due to the fact that there will only be a limited number of kill assessment sensors. Furthermore these sensors will be imperfect. Also note that the problem does not contain any information concerning the geometries of the weapon-target engagements. In practice it is possible for some of these weapons to incapacitate more than one target depending on the geometry of intercept. Since information is being fed back in the dynamic model one would expect that it will have a better performance than the static one. The dynamic versions of the Target-based and Asset-based problems will be called the *Dynamic Target-Based Weapon-Target Allocation Problem* and the *Dynamic Asset-based Weapon-Target Allocation Problem* respectively.

One may ask why consider the Target-Based problem if it is a special case of the Asset-Based problem. The reason is that the Asset-Based problem requires more information (the targeted defense assets) for its formulation than the Target-Based problem. If this information is available, then the Asset-Based formulation is more appropriate. However, if it is not available, then the defense has to use the Target-Based formulation.

Target-Based objectives lead to *subtractive* defense strategies. In other words the defense tries to kill as many of the most lethal targets as possible, or at least the most valuable ones. On the other hand Asset-Based objectives lead to *preferential* defense strategies. In such strategies the defense decides which of its assets should be saved and concentrates all of its weapons on saving these assets. In order to do this however, some of the assets must be sacrificed (i.e left completely undefended). Note that by directing multiple targets at an asset, the offense is in effect trying to make a subtractive defense useless. This is because in a subtractive defense it is likely that at least one of the targets aimed at an asset will get through, and that the asset will almost surely be destroyed. On the other hand, a preferential defense requires much more information because the defense has to know the point of impact of each target. If this information is not available, then the best that the defense can do is to use a subtractive defense. Therefore an understanding of both the Target-Based and Asset-Based problems is needed in order to produce the best defense possible.

In the early stages of an attack, the defense may have very limited knowledge of the trajectories of the targets. It assigns values to the targets based on factors such as target type, probable point of impact, etc. It then assigns weapons to targets with the objective of minimizing the total expected value of the surviving targets after all weapon-target engagements.

In the later stages of the attack the defense may be able to predict the impact of the targets with high probability. These points of impact will be the assets of the defense which will consist of military installations, population centers, Command and Control ($C^2$) nodes, weapon farms, harbors, etc. In order to model the accuracy and reliability of the targets, we will include a parameter called a lethality probability for each target. The lethality probability of a target-asset pair is the probability that the target destroys the asset to which it is aimed if it is not engaged by any of the defense's weapons. This probability will depend on the accuracy of the target as well as the nature (i.e. hardness) of the asset. Since the lethality probabilities of the targets will typically be close to unity, the only way to effectively save an asset is to destroy all of the targets aimed for it. Therefore, for this situation a more appropriate objective is as follows. Assign values to each of the assets and assign weapons to targets so as to maximize the total expected value of the surviving assets after all weapon-target engagements and after all target impacts. Note that different assets will have inherently different values. The value of an asset will depend on the importance of the asset to the defense. In order to save a particular asset, all of the targets aimed for the asset must be engaged, otherwise the asset will be destroyed by the targets which are not engaged.

The efficient solution of the Weapon-Target assignment problem is of great interest to the military. The reason for this is that, in an engagement with the enemy, the problem must be solved in real time. The enormous combinatorial complexity of the problem implies that, even with the supercomputers available today, optimal solutions cannot be obtained in real-time. One must therefore develop good heuristics for solving the problem. To provide good heuristics one must have a thorough understanding of the properties of the problem and its solution. Our intent is to provide properties which will be of use to those who need to provide heuristics.

The main properties of the weapon-target allocation problems can be summarized as follows:

**NP-Complete:** Simple versions of the Target-Based problem have been shown to be NP-Complete [2]. This basically means that there are no efficient methods for finding the optimal solution[2]; one must essentially resort to complete enumeration of all possible allocations. This is an important property since it implies that one should look at even simpler versions to gain insight. This insight can then be used to provide heuristics for the more general problem. Note that more general versions, e.g. the Asset-Based problem, will also be NP-Complete.

**Discrete:** The feasible solutions of these problems are restricted to be integral since only an integral number of weapons can be assigned to a target. Integer programming problems are, in general, difficult to solve.

**Dynamic:** In the case of the dynamic problem one must decide when and to which of the targets each weapon must be assigned. The number of possible allocations therefore grows (exponentially) with the number of time periods. This increases the computational complexity of the problem. However, as we shall see, this "look-shoot-look-shoot..." type of strategy can significantly improve defensive effectiveness.

**Nonlinear:** As we have mentioned before, the objective function of the problems are nonlinear. In some versions it is convex, while in the more general version it is neither convex nor concave.

**Stochastic:** The problems to be considered are stochastic in nature. The task of evaluating the value of an assignment grows with the number of possible outcomes. In the

[2]See Lewis and Papadimitriou [3] for the definition of an NP-Complete problem.

dynamic version of the problem the number of outcomes grows exponentially with the number of time periods. Therefore, for large-scale problems, the task of simply evaluating the value of an assignment may not be possible in practice. This increases the difficulty of the problem since approximations must be used.

**Large-Scale:** The main application of the problems to be considered is that of military defense. For such problems the number of weapons, assets, and targets is enormous. This implies that enumeration techniques are impractical.

These properties of the problem rule out any hope of obtaining efficient optimal algorithms. The purpose of this thesis is therefore to deduce properties of the problem as well as its solution which will be useful in providing good heuristics. Wherever it is possible, we will provide rigorous arguments. Wherever it is appropriate we will provide simple examples and computational results.

## 1.2 Literature Survey

The Weapon-Target assignment problem is an important problem in military defense. Some of the papers that have been written on the subject will be briefly summarized in this section.

In [4], denBroeder et al. consider the special case of the Static Target-based problem in which the kill probability of a weapon-target pair is independent of the weapon (i.e a single class of weapons). They present an optimal algorithm for solving this version of the problem. This algorithm, which is usually referred to as the Maximum Marginal Return (MMR) algorithm in the literature, will be discussed in more detail in chapter 2. Kattar implemented this algorithm and presents some numerical results in [5].

Matlin [6] provides a review of the literature on weapon-target allocation problems. Several references are given and are classified by the model under consideration. Eckler and Burr [7] also give a review of the material on weapons allocation problems. Besides giving references, they summarize different mathematical models and provide some analysis. However, in these studies, very little emphasis is given to the dynamic allocation of weapons which is the main focus of our research.

A major result, obtained by Lloyd and Witsenhausen [2], is that the Static Target-based problem is NP-Complete. What this means is that the computation time of any optimal algorithm for the problem will grow exponentially with the size of the problem. Since this version of the problem is a special case of all the other versions being considered, we can conclude that the other versions are also NP-Complete.

In [8] Soland considers the dynamic version of the Asset-based problem under the assumptions that there is a single asset and that at each stage the kill probability is the same for all weapon-target pairs. He uses stochastic dynamic programming and provides some numerical results. It can be shown that the single-asset, dynamic asset-based problem can be formulated as a dynamic target-based problem with unit-valued targets. Hence, our results for the dynamic target-based problem (chapter 3) can be applied to the problem studied in Soland's paper.

In [9] Burr et al take a different approach to the weapon-target allocation problem. Instead of fixing the number of defensive weapons and minimizing the amount of damage caused by the offense's weapons, they minimize the number of defensive weapons needed by the defense to provide a given level of defense (i.e. an upper bound on the damage caused by the offensive weapons).

A group at Alphatech Inc., under the leadership of Dr. D. A. Castañon, has examined both target-based and asset-based problems in the context of the Strategic Defense System. Their recent reports, although unclassified, are restricted and the author did not have access to these documents. On the other hand, personal communication with Dr. Castañon [10] ensured that no serious duplication of effort and results occurred.

In conclusion, we have found that the open literature on the dynamic versions of the target-based and asset-based problems is scant. Furthermore, the literature which addresses the dynamic problem contains few analytical results because of the difficulty of the problem. Computational results are also limited because most of these are restricted. Our research will focus on the dynamic versions of the target-based and asset-based problems. We will provide both analytical and computational results. Since our research is unrestricted, it can serve as a starting point for those interested in this line of research.

## 1.3 Contributions of Thesis

In Chapter 2 we will present the Static Target-Based WTA Problem. This problem has been shown, by Lloyd and Witsenhausen [2], to be NP-Complete. We have obtained an optimal local search algorithm for the case of a single class of weapons. This algorithm may be preferable to other algorithms for solving this version of the problem because it is easily parallelizable.

In Chapter 3 we will present the Dynamic Target-Based WTA Problem. This problem is also NP-Complete so we looked at some special cases. For the case of identical targets and a uniform kill probability that is independent of the stages, we obtained analytical solutions for the case of two targets and in the limit as the number of targets goes to infinity. These, as well as numerical results, were then used to compare dynamic and static strategies. This comparison is important since it shows that there is a significant performance advantage in using a dynamic strategy instead of a static one. Under suitable assumptions we will show that, approximately half as many weapons are required for a dynamic strategy to obtain the same performance as a static one. We also obtained an analytical solution in the limit as the number of targets goes to infinity under the assumptions of identical targets and stage dependent (but weapon and target independent) kill probabilities. This provides us with a good approximation for problems with many targets. It can be used to investigate how the solution changes as the kill probability in each stage varies. We also looked at the case of weapon independent (but stage and target dependent) kill probabilities. Under these assumptions, the problem is still difficult because multiple local optima may exist. However, if the number of weapons to be used in each stage is fixed, then we can show that a greedy algorithm is optimal for the case of two targets. The key contribution of this chapter is our asymptotic result for the case of unit valued targets and stage dependent kill probabilities. We can conclude that, in general, the performance of the dynamic strategy is significantly greater than that of the static strategy. Furthermore, a good heuristic for the dynamic strategy is to use a MMR type algorithm.

In Chapter 4 we will present the Static Asset-Based WTA Problem. We will assume that the kill probabilities are independent of the weapons (since the problem is NP-Complete otherwise). The question as to whether or not the problem under the assumption of weapon independent kill probabilities is NP-Complete or not is still open. Under this assumption we have obtained a sub-optimal algorithm which produced near-optimal solutions for the problems on which it was applied. The algorithm also produces an upper bound on the optimal value for the problem. We believe that this is the best available algorithm for solving this version of the Static Asset-Based problem. The algorithm was used to obtain several numerical and sensitivity analysis results. From these results we will conclude that the optimal value for the problem is very sensitive to the kill probabilities but insensitive to the lethality probabilities. We will also find that, as the offense increases the number of targets, the defense must increase the number of defense weapons at a greater rate in order to maintain the same level of defense.

In Chapter 5 we will present the Dynamic Asset-Based WTA Problem. We will assume that the kill probability of a weapon-target pair is dependent solely on the asset to which the target is aimed as well as the stage number. This assumption was made for two reasons, (a) to reduce the dimensionality of the problem and (b) the problem is NP-Complete otherwise. Note that even under these assumptions we believe that the problem is still NP-Complete. Our belief is based on the fact that to evaluate the value of a first stage assignment may require an exponential number of operations. We have obtained a sub-optimal algorithm for this version of the problem. A method is also provided for obtaining an upper bound on the optimal value. This algorithm is unique in that it approximates the cost-to-go function in the present stage rather than approximating the cost function for the last stage. Computational results show that the algorithm performs well. The key contribution of this chapter is the proposed heuristic. We can use our results to conclude that the dynamic strategy can offer a significant increase in performance over the static strategy.

# Chapter 2

# The Static Target-Based Problem

In this chapter we will present the static version of the Target-Based WTA problem. This problem has been well studied in the literature. It has been shown by Lloyd and Witsenhausen [2] to be an NP-Complete problem in general. Therefore only sub-optimal algorithms have been proposed for its solution. In the case of a single class of weapons an optimal algorithm has been proposed by denBroeder et al. [4]. The material presented in this chapter is essential for a complete understanding of the results in later chapters.

This chapter is essentially a review of the literature on the Target-B. sed problem. In section 2.1 we will give a mathematical statement of the problem. In section 2.2 we will consider the case of a single class of weapons. In this case the kill probabilities are solely dependent on the targets. The optimal algorithm of denBroeder et al. [4] will be presented in this section. We will also present a new optimal local search algorithm for solving this special case of the problem. In section 2.3 we will present a network flow formulation of the problem under the more general assumption that the kill probability of each weapon-target pair is either zero or, if non-zero, it is solely target dependent. A network flow approach has the advantage that several algorithms already exist for network flow optimization problems. Also in section 2.3 we will present a network flow formulation of the problem under the sole restriction that at most one weapon can be assigned to each target. Several algorithms are available for solving such problems. In section 2.4 we will present a method for obtaining a lower bound on the optimal cost of the problem. This lower bound is obtained by relaxing the constraint that weapons must be assigned in integral quantities. Nonlinear programming techniques can then be used to solve this relaxed problem. One such method is to find and solve the dual of the problem. In 2.5 we will present our conclusions.

## 2.1 Problem Definition

In this version of the problem, the offense launches its weapons (the targets of the defense) at the defense's assets. The defense assigns values[1] to these targets based on the predicted target type, the value of the predicted point of impact and other relevant factors. The defense has weapons which can be used to engage these targets before they impact. A one-to-one kill probability is assigned to each weapon-target pair. This is the probability that the weapon destroys the target if it is assigned to it and reflects such factors as the weapon type, the time and geometry of intercept, the characteristics of the engagement of the specific weapon-target pair and other relevant factors. Therefore, in general, the kill probability of a particular weapon-target pair will be different to the kill probabilities of all other weapon-target pairs. The objective of the defense is to assign its weapons to the targets so as to minimize the expected total value of the surviving targets. Note that in the optimal assignment some high valued targets may be engaged by more than one weapons while others (with low values) may not be engaged by any weapons.

In this version of the problem all weapons are assigned and fired simultaneously. Since there is no time dependence, we will call this a static problem. We will also assume that the state of the targets (survived or destroyed) is observed after all weapons have been fired. In other words there is no feedback of information. This assumption will be valid in cases in which the defense has only a single opportunity to engage the targets. This would occur in conflicts in which the flight duration of the targets is short.

We will also assume that the engagement of a weapon-target pair is independent of all other weapons and targets. In practice this assumption may not hold for all engagements because targets near a weapon-target interception will be affected by the debris of the

[1] Note that it would be more appropriate to call these target costs but we wish to conform to the notation used in the literature.

explosion. However, the problem is very difficult without this assumption because one must then include the geometry of the problem.

The following notation will be used in the mathematical definition of the Static Target-Based problem. The definitions of all additional notation may be found in Appendix A.

$N \stackrel{\text{def}}{=}$ the number of targets (offense weapons),

$M \stackrel{\text{def}}{=}$ the number of defense weapons,

$V_i \stackrel{\text{def}}{=}$ the value of target $i$, $\quad i = 1, 2, \ldots, N$,

$p_{ij} \stackrel{\text{def}}{=}$ the probability that weapon $j$ destroys target $i$ if it is assigned to it, $i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, M$.

The decision variables will be denoted by:

$$x_{ij} = \begin{cases} 1 & \text{if weapon } j \text{ is assigned to target } i, \\ 0 & \text{otherwise.} \end{cases}$$

The probability that target $i$ is not destroyed by weapon $j$ is given by $(1-p_{ij})^{x_{ij}}$. Therefore, since it was assumed that the engagement of a target by a weapon is independent of all other targets and weapons, the probability that target $i$ survives after all weapons have been fired is given by $\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}}$. The problem is therefore given as follows.

---

**Problem 2.1** *The Static Target-Based WTA problem (STB) can be stated as:*

$$\min_{\{x_{ij} \in \{0,1\}\}} F = \sum_{i=1}^{N} V_i \prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}},$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, 2, \ldots, M.$$

---

The objective function, $F : \{0,1\}^{NM} \longrightarrow \Re$, is the total expected value of the surviving targets. We will show that this function is convex.[2] The constraint is due to the fact that each weapon must be assigned to exactly one target.

[2] Note that convex functions are defined in convex sets (see definition A.3.) The set in which $F$ is defined is not convex so it is incorrect to discuss the convexity of this function. In this context, what we really mean is that if we relax the integrality constraint (i.e allow $0 \le x_{ij} \le 1$) then the function $F$ is convex in this set.

**Theorem 2.1** *If we relax the integrality constraint and allow $0 \leq x_{ij} \leq 1$ then the function $F : [0,1]^{NM} \longrightarrow \Re$, as defined in problem 2.1, is convex.*

*Proof:* Since the sum of convex functions is convex then we only need to show that the function $\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}}$ is convex for each index $i$. Let us drop the subscript $i$ and prove that the function

$$g(\vec{x}) \equiv \prod_{j=1}^{M}(1-p_j)^{x_j}$$

is convex in the set $[0,1]^M$. Pick $\vec{y} \in [0,1]^M$ and $\vec{z} \in [0,1]^M$ and let $\lambda \in [0,1]$. Define

$$\phi \equiv g(\vec{y})/g(\vec{z}).$$

We have

$$\lambda g(\vec{y}) + (1-\lambda)g(\vec{z}) = g(\vec{z})[\lambda\phi + (1-\lambda)],$$

and we also have

$$g(\lambda\vec{y} + (1-\lambda)\vec{z}) = g(\vec{z})[\phi^{\lambda}].$$

The function $\phi^{\lambda}$ is a convex function of $\lambda$. Furthermore the function $\lambda\phi + (1-\lambda)$ for $\lambda \in [0,1]$ is a chord of the function $\phi^{\lambda}$. Therefore

$$\phi^{\lambda} \leq \lambda\phi + (1-\lambda), \qquad \lambda \in [0,1].$$

Since $g(\vec{z}) \geq 0$ we can conclude that

$$\lambda g(\vec{y}) + (1-\lambda)g(\vec{z}) \geq g(\lambda\vec{y} + (1-\lambda)\vec{z}).$$

This implies that the function $g(\vec{x})$ is convex which implies that the function $F$ is convex.
■

Figure 2.1 contains a pictorial representation of the problem. The circles on the left represent the weapons while those on t' : right represent the targets. For each weapon-target pair with non-zero kill probability an arrow is drawn from the weapon to the target and labelled with the kill probability of the pair. Each of the targets is labelled with its

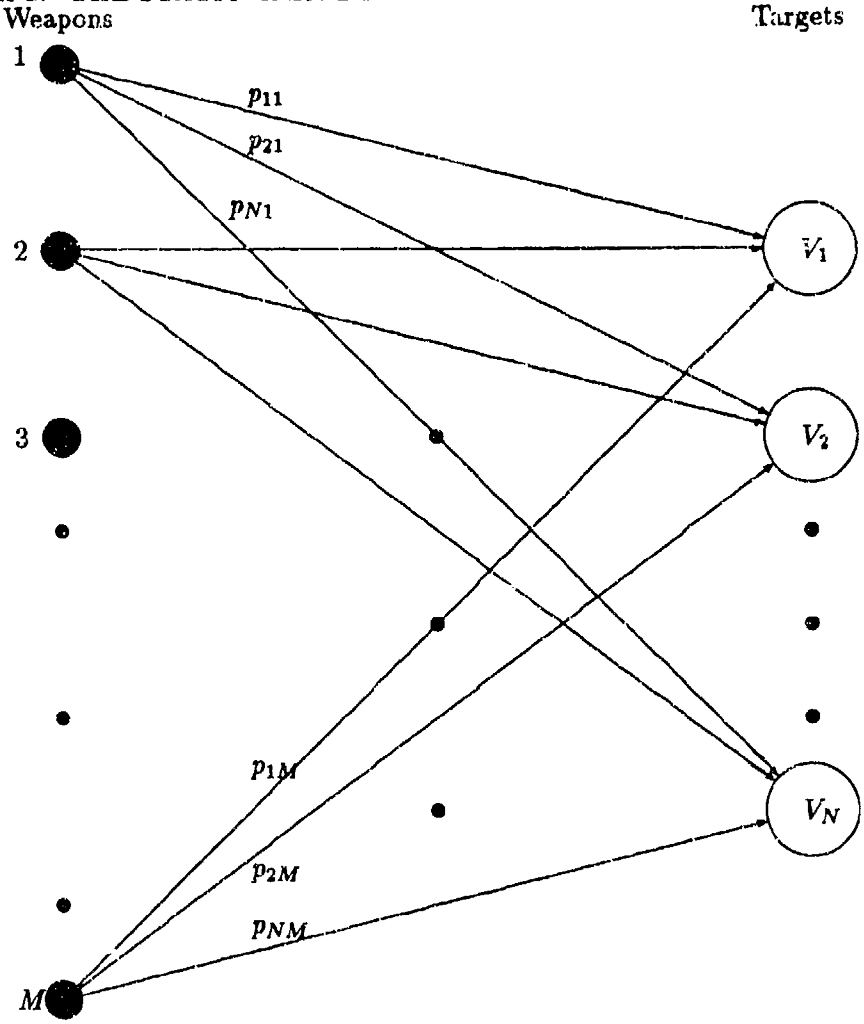


Figure 2.1: Representation of a Static Target-Based Problem

value. Note that a close inspection of such a graph can lead to the discovery of any special structure. This special structure can then be exploited to decrease the computational time or to increase the quality of the solution obtained by a heuristic.

Problem 2.1 has been proven by LLoyd and Witsenhausen [2] to be NP-Complete. Basically what this means is that, if a polynomial time algorithm exists for solving this problem then one also exists for solving many other difficult problems such as the travelling salesman problem. At present, the belief is that polynomial time algorithms do not exist for this class of problems. Note that, although the set of feasible solutions is finite, the number of them, $N^M$, is so large that complete enumeration is not a practical option.

We will see in the next section that if we assume that the kill probabilities do not depend on the weapons (i.e. we have a single class of weapons) then the resulting problem can be solved by a polynomial time algorithm. This implies that the basic difficulty of the problem stems from the fact that there are multiple types of weapons. The problem is also difficult because of the non-linearity of the objective function.

The complexity of the problem suggests that one should rely on sub-optimal algorithms. We will look at optimal algorithms for special cases of the problem. These may then be used as heuristics for the general problem 2.1.

## 2.2 A Single Class of Weapons

In this section we will present two optimal algorithms for solving problem 2.1 under the additional assumption that the kill probabilities are independent of the weapons, i.e. $p_{ij} = p_i$. This assumption is valid if the defense has a single type of weapon and all weapons are located in the same area so that the geometry and time of intercept is the same for all of them.[3] Even if the assumption is not valid, the results of this section can easily be modified to provide a heuristic for the more general version of the problem. Under this assumption, the subscript $j$ can be dropped and we can use the following notation:

$p_i \stackrel{\text{def}}{=}$ the kill probability of a weapon on target $i$, $\quad i = 1, 2 \ldots, N$,

$x_i \stackrel{\text{def}}{=}$ the number of weapons assigned to target $i$, (the decision variable),

$\vec{x} \stackrel{\text{def}}{=} [x_1, \ldots, x_N]^T$,

$Z_+^N \stackrel{\text{def}}{=}$ the set of ordered $N$-tuples of non-negative integers.

[3] Such a grouping of weapons is called a *weapon cluster* in the literature.

We can use these simplifications in the notation to restate the problem.

---

**Problem 2.2** *The Single Weapon Class, Static Target-Based Problem (SSTB) can be stated as:*

$$\min_{\vec{x}\in Z_+^N} F(\vec{x}) = \sum_{i=1}^{N} V_i(1-p_i)^{x_i},$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_i = M.$$

---

Note that the objective function, $F : Z_+^N \longrightarrow \Re$, is convex and separable. These properties reduce the difficulty of the problem.

There are two algorithms that are guaranteed to find the optimal solution to problem 2.2. The first one, due to denBroeder et al. [4] is called the *Maximum Marginal Return* (MMR) Algorithm. This is a polynomial time algorithm with a computational complexity of $O(N + M\log N)$. This implies that problem 2.2 is polynomial time solvable. The second algorithm, due to us, starts from a feasible solution and locally searches for a better solution. We will therefore call this a *Local Search* (LS) algorithm. The latter algorithm has the advantage that, if small changes in the problem parameters are made then little additional work is required to obtain the new optimal solution.

### 2.2.1 The Maximum Marginal Return Algorithm

In the Maximum Marginal Return algorithm, weapons are assigned sequentially to the target for which the reduction in the objective cost is maximum. The algorithm terminates after all weapons have been assigned. The Pidgin Algol code for the algorithm is given in figure 2.2.

The marginal return of adding an additional weapon to a target is represented by $\Delta_i$. The target with the maximum return is found and the number of weapons assigned to this target is increased by one. The marginal return of this target is updated and the procedure

```
procedure MMR
begin
    x := [0,...,0]^T;
    for i = 1:N do Δ_i := V_i p_i
    for j = 1:M do
    begin
        Let k be such that Δ_k = max_i{Δ_i};
        x_k := x_k + 1;
        Δ_k := V_k p_k (1 - p_k)^{x_k};
    end
end
```

Figure 2.2: The Maximum Marginal Return Algorithm applied to the Single Weapon Class, Static Target-Based problem.

is repeated until all weapons have been assigned. Note that the marginal return if $x_i$ is increased by one is given by

$$\Delta_i = V_i[(1-p_i)^{x_i} - (1-p_i)^{x_i+1}] = V_i p_i (1-p_i)^{x_i}.$$

Since initially $x_i = 0$ for all targets $i$ then the initial marginal returns are given by $\Delta_i = V_i p_i$.

**Theorem 2.2** *The assignment produced by the MMR algorithm is optimal for problem 2.2.*

*Proof:* Since the functions $V_i(1-p_i)^{x_i}$ are convex then problem 2.2 satisfies the conditions that are required in order to apply theorem B.1. Theorem B.1 can therefore be applied to prove optimality. The application of the theorem is straightforward. ■

Note that the assignment of a weapon to a target reduces the target's probability of survival. One way to reflect this is to reduce the target's value to its expected surviving value. This is precisely what the MMR algorithm does. It assigns the first weapon to the target for which the resulting reduction in value is maximal. The value of this target is reduced to its expected surviving value and the process is repeated until all weapons are assigned.

The MMR algorithm is extremely simple as well as fast. The marginal returns can

```
procedure LS
begin
    let x be any feasible assignment;
    for i = 1:N do Δ_i := V_i p_i (1 - p_i)^{x_i}
    modified := TRUE;
    while (modified == TRUE) do
    begin
        modified := FALSE;
        for i = 1:N do
        for j = 1:N do
        begin
            if Δ_i/(1 - p_i) < Δ_j and x_i > 0 then
            begin
                x_i := x_i - 1;
                x_j := x_j + 1;
                modified := TRUE;
            end
            elseif Δ_j/(1 - p_j) < Δ_i and x_j > 0 then
            begin
                x_j := x_j - 1;
                x_i := x_i + 1;
                modified := TRUE;
            end
        end
    end
end
```

Figure 2.3: The Local Search Algorithm

initially be numerically ordered in $O(N)$ time. After each iteration, the updated marginal return must be re-inserted into the list. This can be done in $O(\log N)$ time. Since the number of iterations is $M$, the computational complexity of the algorithm is $O(N + M \log N)$.

### 2.2.2 The Local Search Algorithm

The Local Search Algorithm starts with any feasible solution to the problem. It then searches for a pair of targets such that the removal of a weapon from one of the targets and the addition of this weapon to the other target reduces the cost. This process is repeated until no more reductions can be made. This is called a local search algorithm since it searches for a descent direction in the neighborhood of the present solution. The code for this algorithm is given in figure 2.3.

The marginal returns $\Delta_i$ are first computed for each target $i$. For each pair of targets $i$ and $j$ with $x_i > 0$ we check to see if removing a weapon from target $i$ and adding it to target $j$ decreases the objective cost. Note that the increase in cost if a weapon is removed from target $i$ is given by $V_i(1-p_i)^{x_i-1}$. Since $\Delta_i = V_i(1-p_i)^{x_i}$ then the increase in cost can also be written as $\Delta_i/(1-p_i)$. Note that for each iteration in which the solution is updated the objective cost is strictly decreased. Since the number of feasible solutions is finite then the algorithm must terminate after a finite number of iterations.

**Theorem 2.3** *The assignment produced by the LS algorithm is optimal for problem 2.2.*

*Proof:* Let $\vec{x}^*$ denote the vector of optimal assignments and let $\vec{x}$ denote the assignment obtained on termination of the LS algorithm. Let $F(\vec{x}^*)$ and $F(\vec{x})$ denote the corresponding costs. Let us assume that $F(\vec{x}^*) < F(\vec{x})$ (i.e. that the solution produced by the LS algorithm is not optimal). This implies that there exist targets $i$ and $k$ with the property that $x_i^* > x_i$ and $x_k^* < x_k$. By the nature of the algorithm it must be that

$$V_k p_k (1-p_k)^{x_k-1} \geq V_i p_i (1-p_i)^{x_i}.$$

We therefore have

$$V_k p_k (1-p_k)^{x_k^*} \geq V_k p_k (1-p_k)^{x_k-1} \geq V_i p_i (1-p_i)^{x_i} \geq V_i p_i (1-p_i)^{x_i^*-1} \qquad (2.1)$$

Note that equality must hold throughout in the expression 2.1 otherwise the assignment $\vec{x}^*$, which was assumed to be optimal, could be improved by removing a weapon from target $i$ and assigning it to target $k$ leading to a contradiction. On the other hand, if equality holds throughout then, if a weapon is removed from target $i$ and assigned to target $k$ in assignment $\vec{x}^*$ then the resultant assignment, which we will call $\vec{z}^*$, is also optimal. Note that the assignment $\vec{z}^*$ is closer to assignment $\vec{x}$ (i.e. $|\vec{x} - \vec{x}^*| > |\vec{x} - \vec{z}^*|$). We can now repeat the process to get another optimal solution which is even closer to assignment $\vec{x}$ than is $\vec{z}^*$. After a finite number of repetitions of this process we will obtain the result that $\vec{x}$ is also an optimal assignment. ∎

One desirable property of the LS algorithm is that several of the descent iterations can be done in parallel. For example, suppose that two processors P1 and P2 were available to solve the problem. Assume (for simplicity) that $N$ is even. Let us assign P1 to targets 1 to $N/2$, and P2 to targets $N/2+1$ to $N$. Each processor will be allowed to assign $M/2$ weapons. Starting from any feasible solution, P1 can execute the LS algorithm on its targets while P2 can execute it on its targets. Let $\tilde{x}$ denote the solution after P1 and P2 have each executed the LS algorithm on their targets. Let $\tilde{\Delta}$ denote the corresponding set of marginal returns

$$\Delta_i = V_i p_i (1 - p_i)^{x_i}.$$

Note that if

$$\min_{1 \le i \le N/2} \left\{ \frac{\Delta_i}{1 - p_i} \right\} \ge \max_{N/2 < j \le N} \{\Delta_j\}, \tag{2.2}$$

and

$$\min_{N/2 < j \le N} \left\{ \frac{\Delta_j}{1 - p_j} \right\} \ge \max_{1 \ge i \le N/2} \{\Delta_i\}, \tag{2.3}$$

then the solution $\tilde{x}$ cannot be improved by a pairwise swap hence it is optimal. If, on the other hand, the inequality 2.2 does not hold, then the solution can be improved by removing weapons from one or more of the targets assigned to P1 and adding them to the targets assigned to P2. Similarly, if inequality 2.3 does not hold, then the solution can be improved by removing weapons from one or more of the targets assigned to P2 and adding them to the targets assigned to P1. The algorithm can then be executed on P1 and P2. This process can be repeated until the inequalities 2.2 and 2.3 both hold. Note that the problem solved by each processor is essentially half of the size of the original problem. Also note that each of the processors has to execute the LS algorithm on half of the targets. Instead of using one processor to do this we can again split the problem (into two subproblems of $N/4$ targets each) and use two processors to execute the LS algorithm on the $N/2$ targets. In this manner the method can be extended to more than two processors.

## 2.3 Network Flow Formulations of Special Cases

In this ection we will formulate special versions of problem 2.1 as Network Flow Optimization problems. By doing this we can make use of several efficient algorithms which are already available for Network Flow Optimization problems. We will find that this approach works well for the case of the Static Target-Based problem. However, Network Flow formulations are not possible for the Asset-Based problem or for the dynamic versions of the Target-Based and Asset-Based problems.

### 2.3.1 Weapons with Limited Target Coverage

In the previous section we had assumed that the kill probability is independent of the weapons. In this section we will assume that for each weapon-target pair the weapon can either be assigned to the target or it cannot be assigned to the target (i.e. each weapon can only reach some of the targets). If it can be assigned to the target then we will assume that the kill probability of the pair is only dependent on the target. In other words we are assuming that the kill probability of a weapon-target pair is either 0 or some target dependent value $p_i$ (i.e $p_{ij} \in \{0, p_i\}$). This assumption is appropriate for the case in which the weapons are geographically distributed. In such a case each weapon may only be able to reach a subset of the targets. The kill probability of a weapon on a target which it cannot reach can be set to zero. If this is done then the weapon will not be assigned to the targets it cannot reach. Let $C_i$ denote the set of weapons which can reach target $i$ (i.e the set of weapons $j$ such that $p_{ij} > 0$).

---

**Problem 2.3** *The Limited Target Coverage problem can be stated as:*

$$\min_{\{x_{ij} \in \{0,1\}\}} F = \sum_{i=1}^{N} V_i (1 - p_i)^{\sum_{j \in C_i} x_{ij}},$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, 2, \ldots, M.$$

---

Note that the survival probability of each target depends on the sum of the weapons aimed for it. It is this fact which makes a network formulation of the problem possible.

We will define a network for this problem as follows. For each weapon and each target we will include a node (see figure 2.4). For each weapon-target pair we will include an arc between these nodes if the kill probability of the pair is non-zero. Denote this set of arcs by $A$. Each of the weapon nodes will be a supply node with a supply of 1. We will also include a sink node $s$. The sink node will have a demand of $M$. For each target node we will

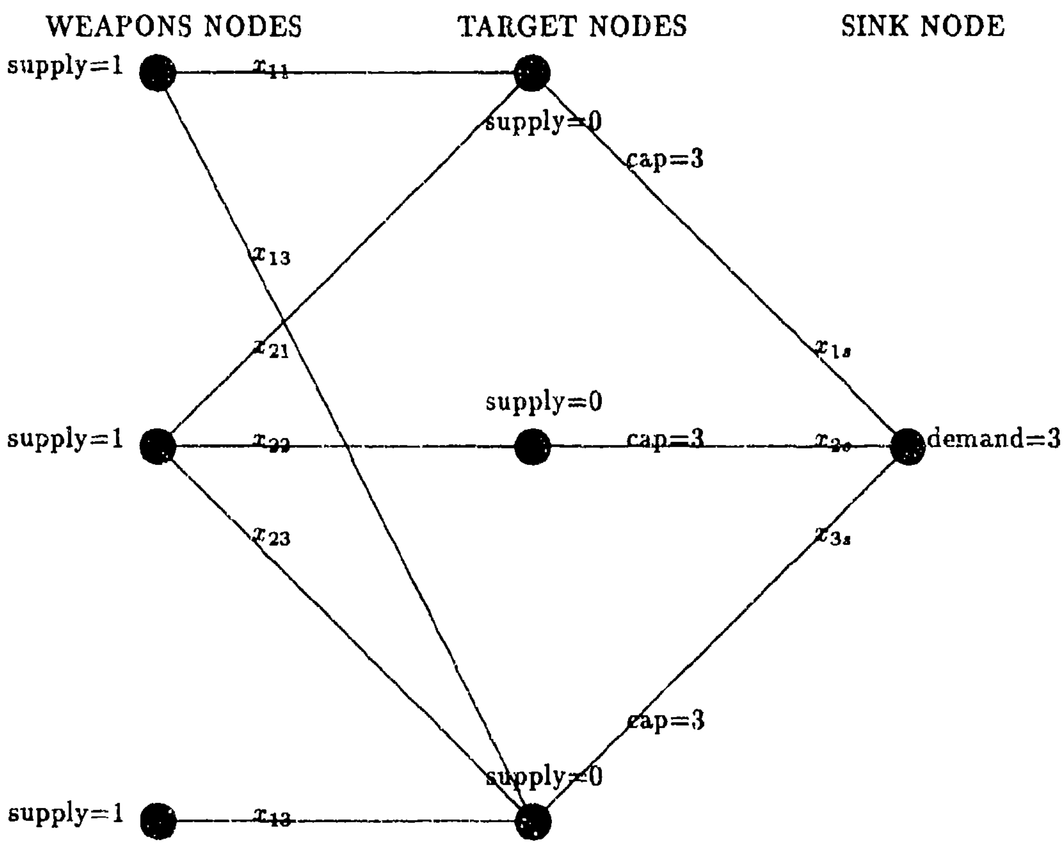


Figure 2.4. Example of the Network Flow representation of a Static Target-Based problem with 3 weapons and 3 targets. Arcs are drawn between a weapon-target pair only if the kill probability of the weapon on the target is non-zero. These arcs have a capacity of unity.

include an arc from the target node to the sink node $s$. The flow on the arc from weapon node $i$ to target node $j$ will be denoted by $x_{ij}$.[4] The flow on the arc from target node $j$ to the sink node $s$ will be denoted by $x_{js}$. All arcs from weapon nodes to target nodes will have a lower bound of 0 and an upper bound of 1. All arcs from the target nodes to the sink node $s$ will have a lower bound of 0 and an upper bound of $M$. There will be no cost associated with flow on arcs from the weapon nodes to the target nodes. The cost of having a flow of $x_{js}$ on the arc from target node $j$ to the sink node will be denoted by $F_{js}(x_{js})$.

The functions $F_{js} : [0, M] \longrightarrow \Re$ are given by[5]:

$$F_{js}(x) = V_j[(1-p_j)^{\lfloor x \rfloor} + (x - \lfloor x \rfloor)((1-p_j)^{\lceil x \rceil} - (1-p_j)^{\lfloor x \rfloor})], \qquad 0 \leq x \leq M.$$

[4]The ordering of the subscripts is different to that used in the definition of problem 2.1. We have changed the ordering to conform to the notation commonly used in graph theory for denoting an ordered arc.

[5]The reader can refer to Appendix A for a definition of the notation $\lfloor x \rfloor$ and $\lceil x \rceil$.

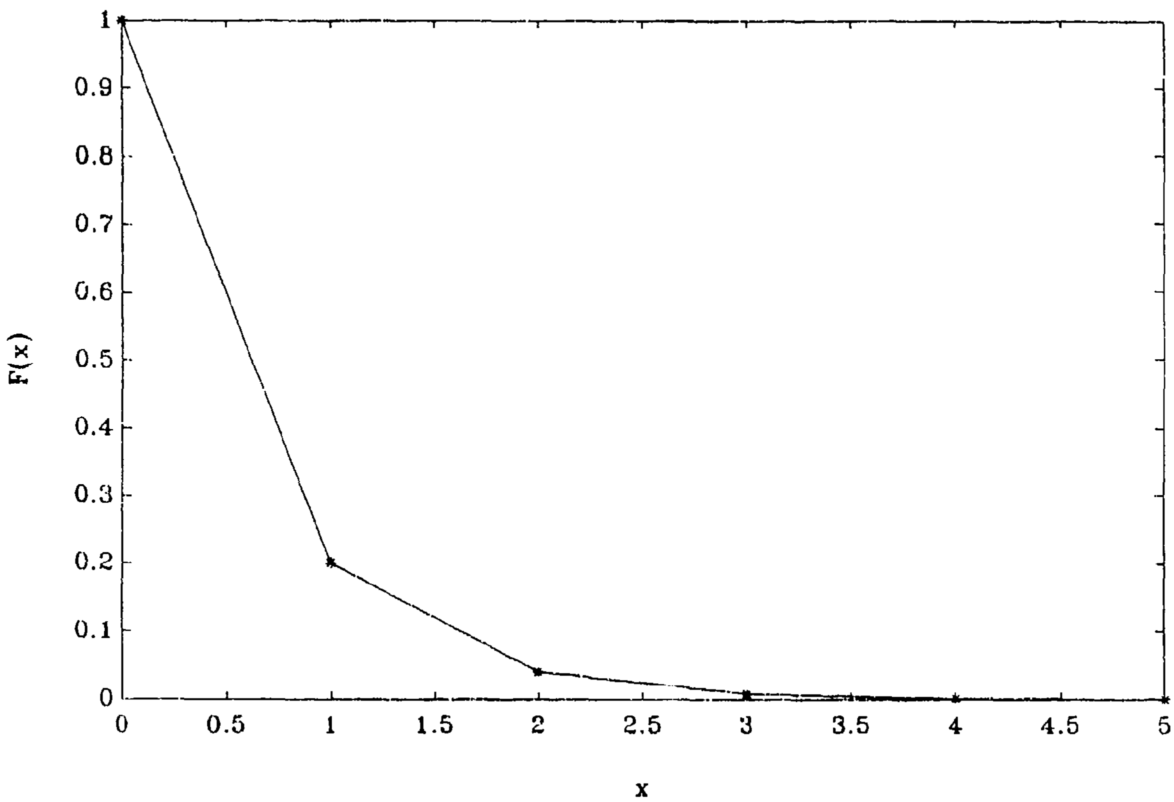


Figure 2.5: An example of the arc flow cost, $F_{js}(x_{js})$, for a target $j$ with value $V_j = 1$ and kill probability $p_j = 0.8$. Function values for non-integral flows are obtained by interpolation of the values for integral flows.

This function can be described in words as follows. If $k \in Z_+$ weapons are assigned to target $j$ then the expected surviving value is given by $V_j(1-p_j)^k$. Note that this is the same as $F_{js}(k)$ which is what we want. However, in the network flow problem the arc flows $x_{js}$ are allowed to have real values. The cost function must therefore be defined for real valued flows. We have done this by letting $F_{js}(x)$ for $0 \leq x \leq M$ be the linear interpolation of the function $F_{js}(x)$ defined for $x = 0, 1, \ldots, M$. For example, consider the case in which $p_j = 0.8$ and $V_j = 1$ for some target node $j$. In figure 2.5 we have plotted the function $V_j(1-p_j)^{x_j}$ for $x_j = 0, 1, 2, 3, 4, 5$ with stars. The linear interpolation of this function is plotted with the solid line. Therefore the solid line represents the function $F_{js}(x_{js})$. We can now define the corresponding network flow optimization problem.

---

**Problem 2.4** *The Convex, Minimum Cost Network Flow problem (CNET) can be stated as:*

$$\min_{\{x_{js} \in [0,M]\}} F = \sum_{j=1}^{N} F_{js}(x_{js}),$$

subject to

$$\begin{array}{lll}
(a) & \sum_{\{j|(i,j)\in A\}} x_{ij} = 1 & i = 1, 2, \ldots, M, \\
(b) & \sum_{\{i|(i,j)\in A\}} x_{ij} = x_{js} & j = 1, 2, \ldots, N, \\
(c) & \sum_{j=1}^{N} x_{js} = M & \\
(d) & 0 \leq x_{ij} \leq 1 & \forall \quad (i,j) \in A \\
(e) & 0 \leq x_{js} \leq M & j = 1, 2, \ldots, N.
\end{array}$$

---

Note that the weapons are indexed by $i$ while the targets are indexed by $j$. The objective function is the sum of the cost of the flow over each arc. Constraint (a) is due to the fact that the weapon nodes are supply nodes with supply 1. Similarly constraint (c) is due to the fact that node $s$ is a sink node with demand $M$. Constraint (b) requires that the total flow into a target node must equal the total flow out of it. Constraint (d) is due to capacity bounds on the arcs from the weapon nodes to the target nodes. It restricts the number of weapons assigned from a weapon node to a target node to one. Constraint (e) is due to the capacity bounds on the arcs from the target nodes to the sink node $s$.

Note that if the flows are restricted to be integral then the network flow problem is identical to problem 2.3. This can be seen as follows. Since the flow is integral then, because of the constraints, exactly one of the arcs out of each weapon node will have a flow of 1. The target node to which this arc belongs is the target to which this weapon would be assigned. Therefore, this implies that the flows $x_{ij}$ are integral. If the flows $x_{ij}$ are integral then the flows $x_{js}$ are also integral. However for integral values of $x_{js}$ the function $F_{js}$ is the same as the expected surviving value of target $j$ which is the function being minimized in problem 2.3.

**Theorem 2.4** *An optimal solution to problem 2.4 exists in which all flows are integral.*

*Proof:* This problem can be transformed into a minimum cost network flow problem with linear arc costs as follows. Instead of a single arc from each target node to the sink node we will include $M$ arcs between each target node and the sink node. Each of these arcs will have a capacity of one. Each of the $M$ arcs from target node $j$ will represent one of the $M$ linear segments of the function $F_{js}$ and will have an arc flow cost equal to the gradient of the linear segment which it represents. The solution of this problem is the same as the solution of the problem CNET. Since this is a linear cost problem and the arc capacities, the supplies and the demands are all integral, then there exists an optimal solution in which all arc flows are integral (see page 239 of [11] for details). ■

We have shown that the solution to the network flow problem 2.4 is integral. Furthermore we have shown that, if the optimal solution to the network flow problem is integral then it is optimal for problem 2.1. Therefore, the optimal values of the variables $\{x_{ij}\}$ of the network flow problem is optimal for problem 2.1. Several algorithms have been proposed for solving the network flow problem 2.4. The reader can refer to [11] for the details of some of these algorithms.

### 2.3.2 Special Case of at most one Weapon per Target

In this section we will consider problem 2.1 under the additional assumption that each target can be assigned at most one weapon. We will show that this can be converted into a Linear Minimum Cost Network Flow Problem. Note that this constraint implicitly assumes that $M \leq N$.

Let us first note that if $x_{ij} \in \{0,1\}$ for $j = 1,2,\ldots,M$, then

$$\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}} = \prod_{j=1}^{M}(1-p_{ij}x_{ij}).$$

Second note that if we also have $\sum_{j=1}^{M} x_{ij} \leq 1$, then

$$\prod_{j=1}^{M}(1-p_{ij}x_{ij}) = 1 - \sum_{j=1}^{M} p_{ij}x_{ij}.$$

Therefore, under the constraint that at most one weapon can be assigned to each target, problem 2.1 can be simplified.

---

**Problem 2.5** *The Linear Cost Network Flow problem (LNET) can be stated as:*

$$\max_{\{x_{ij}\in\{0,1\}\}} F = \sum_{i=1}^{N}\sum_{j=1}^{M} V_i p_{ij} x_{ij},$$

subject to

$$\sum_{i=1}^{N} x_{ij} = 1, \quad j = 1,2,\ldots,M,$$
$$\sum_{j=1}^{M} x_{ij} \leq 1, \quad i = 1,2,\ldots,N.$$

---

The first constraint is due to the fact that each weapon can be assigned to at most one target while the second is the additional constraint that each target can be assigned at most one weapon.

Problem 2.5 is called a *Transportation Problem*, which is a special case of the Linear Minimum Cost Network Flow Problem. If $M = N$ then it is called a *Weighted Bipartite Matching Problem*[6]. These problems have been well studied and many algorithms are available for solving them. Details on these algorithms can be found in [13] and [11]. The paper by Orlin [14] also discusses Network Flow formulations for the weapon to target allocation problem.

We can conclude that, by transforming special cases of the Static Target-Based problem into Network Flow optimization problems we can efficiently solve these special cases. This approach works well for the static version of the Target-Based problem. However, the approach is not easily extendable to more general versions of the weapon to target allocation problem.

## 2.4 Bounds on the Optimal Cost

Since the general problem is NP-Complete, heuristics will have to be used to solve it. It is therefore helpful to have bounds on the optimal cost so that one can have an idea of the performance of these heuristics. The cost of any feasible solution is an upper bound on the optimal cost so we will concentrate on finding a lower bound.

Recall that one of the constraints of problem 2.1 is that the decision variables must be integral. Let us call the problem in which this constraint is relaxed, the Relaxed Static Target-Based Problem. The solution to the Relaxed problem would be the optimal solution to the problem in which the defense was allowed to fire "fractional" weapons at the targets. Note that the optimal cost of the Relaxed problem is a lower bound on the optimal cost of problem 2.1. This is due to the fact that the feasible set of the relaxed problem contains the feasible set of problem 2.1. The Relaxed problem will be solved to obtain a lower bound on the optimal cost of 2.1. Note that the Relaxed problem is easier to solve because nonlinear

[6]The weights for this problem are given by $V_i p_{ij}$.

programming techniques can be applied.

In the relaxed problem the constraint $x_{ij} \in \{0,1\}$ is relaxed to $x_{ij} \in [0,1]$. Furthermore, since $x_{ij} \geq 0$ then the constraint $\sum_{i=1}^{N} x_{ij} = 1$ implies that $x_{ij} \leq 1$. Therefore the constraint $x_{ij} \leq 1$ is not necessary.

---

**Problem 2.6** *The Relaxed Static Target-Based problem (RSTB) can be stated as:*

$$\min_{\{x_{ij} \geq 0\}} F = \sum_{i=1}^{N} V_i \prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}},$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, 2, \ldots, M.$$

---

This is a *Convex Programming Problem*[7]. Since the problem involves minimizing a convex function over a compact, convex set then it can be shown that an optimal solution exists. Furthermore the optimal cost is finite. In the following discussion this problem will also be referred to as the Primal problem.

We will now define the dual of the relaxed problem 2.6. The reader can refer to Appendix A for the definitions of the notation used. It can be shown that the optimal cost of the dual problem is the same as the optimal cost of the primal problem. Therefore, the dual problem can be used to obtain the optimal cost for the relaxed problem which will provide us with a lower bound on the optimal cost of problem 2.1.

Define the matrix $X \in [0,1]^{N \times M}$ by

$$[X]_{ij} \equiv x_{ij}.$$

[7] See [12] for a definition of a Convex Programming problem.

The *Lagrangian function* $L(X,\lambda)$, defined for $X \in \Re^{N\times M}$ and $\lambda \in \Re^M$ for problem 2.6 is given by

$$\begin{aligned} L(X,\lambda) &\equiv \sum_{i=1}^{N} V_i \prod_{j=1}^{M}(1-p_{ij})^{x_{ij}} + \sum_{j=1}^{M}\lambda_j\left[\sum_{i=1}^{N}x_{ij}-1\right], \\ &= \sum_{i=1}^{N}\left(V_i\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}} + \sum_{j=1}^{M}\lambda_j x_{ij}\right) - \sum_{j=1}^{M}\lambda_j. \end{aligned} \tag{2.4}$$

The *dual functional* $q : \Re^M \longrightarrow \Re$ is then given by

$$q(\lambda) = \min_{X\geq 0} L(X,\lambda) \tag{2.5}$$

where we have used the notation $X \geq 0$ to represent $x_{ij} \geq 0$ for $i = 1,2,\ldots,N$ and $j = 1,2,\ldots,M$. Let us derive an explicit expression for $q(\lambda)$.

$$\begin{aligned} q(\lambda) &= \min_{X\geq 0} L(X,\lambda) \\ &= \min_{X\geq 0}\sum_{i=1}^{N}\left(V_i\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}} + \sum_{j=1}^{M}\lambda_j x_{ij}\right) - \sum_{j=1}^{M}\lambda_j \\ &= -\sum_{j=1}^{M}\lambda_j + \sum_{i=1}^{N}\min_{\vec{x}_i\geq 0}\left(V_i\prod_{j=1}^{M}(1-p_{ij})^{x_{ij}} + \sum_{j=1}^{M}\lambda_j x_{ij}\right). \end{aligned}$$

The vector $\vec{x}_i \in \Re^M$ is the $i^{\text{th}}$ row of the matrix $X$. Note that the problem has been simplified into $N$ subproblems. Let us consider one of these subproblems. The subscript $i$ will be dropped to simplify the notation. Each subproblem can be written as

$$\min_{\vec{x}\geq 0} g(\vec{x}) \equiv V\prod_{j=1}^{M}(1-p_j)^{x_j} + \sum_{j=1}^{M}\lambda_j x_j. \tag{2.6}$$

where $g : \Re^m \longrightarrow \Re$ is the objective function. Note that $g$ is also dependent on $\lambda$ and is a convex function of $\vec{x}$. Define

$$\Gamma \equiv \min_{j=1,2,\ldots,M}\left\{\frac{-\lambda_j}{V\ln(1-p_j)}\right\}$$

and

$$\Phi \equiv \prod_{j=1}^{M}(1-p_j)^{x_j}.$$

Note that $\Phi \leq 1$. The derivative of $g$ along the $j^{\text{th}}$ coordinate is given by

$$\frac{dg}{dx_j} = V\Phi\ln(1-p_j) + \lambda_j.$$

We have the following lemma.

**Lemma 2.1** *If we denote the optimal solution to problem 2.6 by $\vec{x}^*$ then $\Gamma \geq 1$ if and only if $\vec{x}^* = \vec{0}$.*

*Proof:* If $\Gamma \geq 1$ then, by the definition of $\Gamma$, we know that for $j = 1, 2, \ldots, M$

$$\frac{-\lambda_j}{V\ln(1-p_j)} \geq 1 \geq \Phi.$$

This implies that $\frac{dg}{dx_j} \geq 0$. If this is the case then, because the function $g$ is strictly convex then it must be that $x_j^* = 0$ for $j = 1, 2, \ldots, M$.

Let us now prove the converse. If $\vec{x}^* = 0$ then $\frac{dg}{dx_j} \geq 0$ for all $j$. We also have $\Phi = 1$. Therefore

$$V\ln(1-p_j) + \lambda_j = V\Phi\ln(1-p_j) + \lambda_j \geq 0.$$

This implies that $\Gamma \geq 1$. □

Let us now assume that $0 \leq \Gamma < 1$. Suppose that $\Gamma = -\lambda_k/(V\ln(1-p_k)$ (i.e the minimum in the definition of $\Gamma$ is achieved for the index $k$). Since $\frac{dg}{dx_j} \geq 0$ at $x_k = x_k^*$ then

$$V\Phi(\vec{x}^*)\ln(1-p_k) + \lambda_k \geq 0,$$

which implies that

$$\Phi(\vec{x}^*) \leq \frac{-\lambda_k}{V\ln(1-p_k)} = \Gamma. \tag{2.7}$$

By Lemma 2.1 there must exist an index $j$ such that $x_j^* > 0$ which implies that

$$V\Phi(\vec{x}^*)\ln(1-p_j) + \lambda_j = 0.$$

$$\Longrightarrow \Phi(\vec{x}^*) = \frac{-\lambda_j}{V\ln(1-p_j)} \geq \Gamma. \tag{2.8}$$

Using the inequalities 2.7 and 2.8 we conclude that $\Phi(\vec{x}^*) = \Gamma$. Therefore we can write

$$\begin{aligned} & \Phi(\vec{x}^*) = \Gamma \\ \Longrightarrow \quad & \prod_{j=1}^{M}(1-p_j)^{x_j^*} = \Gamma \\ \Longrightarrow \quad & \sum_{i=1}^{M} x_j^* \ln(1-p_j) = \ln\Gamma \\ \Longrightarrow \quad & \sum_{i=1}^{M} x_j^* \lambda_j = -V\Phi\ln\Gamma. \end{aligned}$$

Finally we have:

$$g(\vec{x}^*) = V\Gamma(1-\ln\Gamma).$$

If we denote the optimal value of problem 2.6 by $g^*$ then

$$g^*(\lambda) = \begin{cases} V\Gamma(\lambda)(1-\ln\Gamma(\lambda)) & \text{for} \quad 0 \le \Gamma(\lambda) < 1 \\ V & \text{for} \quad \Gamma(\lambda) \ge 1 \end{cases} \tag{2.9}$$

where

$$\Gamma(\lambda) \equiv \min_{j=1,2,\ldots,M} \left\{ \frac{-\lambda_j}{V\ln(1-p_j)} \right\}.$$

The result in 2.9 can now be used to write the dual functional explicitly as

$$q(\lambda) = \sum_{i=1}^{N} g_i^*(\lambda) - \sum_{j=1}^{M} \lambda_j \tag{2.10}$$

where

$$g_i^*(\lambda) = \begin{cases} V_i\Gamma_i(\lambda)(1-\ln\Gamma_i(\lambda)) & \text{for} \quad 0 \le \Gamma_i(\lambda) < 1 \\ V_i & \text{for} \quad \Gamma_i(\lambda) \ge 1 \end{cases}$$

and $\Gamma_i(\lambda)$ is defined as

$$\Gamma_i(\lambda) = \min_{j=1,2,\ldots,M} \left\{ \frac{-\lambda_j}{V_i\ln(1-p_{ij})} \right\}.$$

The *dual* of problem 2.6 is given by

$$\max_{\lambda\ge 0} q(\lambda). \tag{2.11}$$

Note that this is an unconstrained concave maximization problem. It can be shown that for Convex Programming problems

$$F^* = q(\lambda^*) \equiv \max_{\lambda\ge 0} q(\lambda).$$

where $F^*$ is the optimal cost of the primal problem. Therefore $F^*$ is equal to the optimal cost of the dual problem.

The dual functional is concave. Unfortunately it is not differentiable everywhere. However, there are methods, such as subgradient methods, for maximizing non-differentiable concave functions. The advantage in solving the dual problem instead of the primal problem is that the number of variables in the dual problem is $M$, while the number in the primal problem is $NM$. We therefore expect that algorithms for solving the dual problem will be faster than those for the primal problem.

There are three general methods that can be used to solve problem 2.6 *Direct Primal Methods*, *Primal-Dual Methods* and *Direct Dual Methods*. Direct Primal methods produce algorithms which solve for the decision variables $x_{ij}$ directly. Algorithms in this class include Feasible Direction Algorithms, Manifold Suboptimization Methods and Projected Newton Methods (see [12]). The computation time depends on the number of variables, which in this case is $NM$. For the class of problems that we are interested in, this number will be large so that these algorithms are, in general, impractical. Primal-Dual methods solve both the primal problem 2.6 and the dual problem 2.11 simultaneously. This class includes Relaxation Methods which are iterative in nature. One advantage of this method is that the cost of any feasible solution to the dual problem is a lower bound to the optimal cost of the primal problem so that if the iterations are stopped prematurely and the closest primal feasible solution is found, one can obtain a lower bound and check the quality of the solution. Direct dual methods solve the dual problem 2.11. As we have seen above, the optimal cost for this problem is also the optimal cost for the primal problem 2.6. The number of dual variables equals the number of constraints which, for this problem is $M$. Since this is much less than the number of primal variables one would expect faster computation time than direct primal methods. These methods however suffer from the fact that the dual functional is non-differentiable.

The best method to use for finding a lower bound on the optimal cost for problem 2.2 will depend on the class of problems to be solved. For large-scale problems, the direct dual methods will probably be more efficient than direct primal methods. The paper by Pugh [15] considers the use of Lagrange Multiplier methods for the solution of weapon to target allocation problems.

## 2.5 Concluding Remarks

In this chapter we have presented a summary of previous work done on the Static Target-Based WTA problem. We also presented a new algorithm, the Local Search algorithm, for solving the problem under the additional assumption of weapon independent kill probabilities.

We considered the special case in which the kill probability of a weapon-target pair can either be 0 (meaning that the weapon cannot be assigned to the target) or, if it is non-zero, it depends only on the target. In other words each weapon can only reach a subset of the targets. This problem can be solved by first converting it to a Minimum Cost Network Flow Problem with Convex Arc Costs and then using algorithms for that problem.

We next considered the special case with the additional constraint that each target can be assigned at most one weapon. This problem can be shown to be equivalent to a Transportation Problem. Many polynomial time algorithms are available for solving the Transportation Problem.

Note that these problems will have to be solved in real time since the engagement time may be short. One method for solving these problems quickly is to solve them on parallel computers. Therefore, research in parallel algorithms for the problem is necessary. Furthermore, the weapons will be geographically distributed. This suggests the use of distributed computation. This is another area of research that should be investigated.

# Chapter 3

# The Dynamic Target-Based Problem

In this chapter we will consider the dynamic version of the Target-Based WTA Problem. This problem consists of a number of time stages. The defense is allowed to observe the outcomes of all engagements of the previous time stage before assigning and commiting weapons for the present stage. This is called a "shoot-look-shoot-..." strategy since the defense is alternating between shooting its weapons and observing (looking) at the outcomes. Note that in a real conflict it is not possible to divide time into distinct stages as we have done. This is because the offense's weapons do not all arrive simultaneously and, even if they did, the time of each weapon-target engagement will be different. However, without the assumption of distinct time stages, the complexity of the problem is to great to obtain any analytical results and hence any insights.

In section 3.1 we will give a mathematical definition of the problem. In section 3.2 we will consider the effect of stage dependent (but weapon and target independent) kill probabilities on the optimal assignment. We will assume identical targets and provide an analytical solution in the limit as the number of targets goes to infinity. We will find that if the weapon-target ratio is kept fixed, then, in the limit as the number of targets goes to infinity, the problem can be considered as a deterministic one in which the number of targets in each stage is equal to the expected number of targets which survive the previous stage. This simplifies the analysis since a deterministic problem is much easier to solve. In section

3.3 we will consider the special case of identical targets and a uniform kill probability for all weapon-target pairs and all stages. In this section we will provide analytical results for the case of two targets as well as for the limit as the number of targets goes to infinity. These results will then be used to analytically compare dynamic and static strategies. We will obtain the very interesting result that roughly half as many weapons are required for the dynamic strategy to obtain the same performance as that of the static strategy. Section 3.4 contains the special case of weapon independent kill probabilities. We will demonstrate that this problem is difficult because multiple minima may exist. We will provide an algorithm for finding one of these local minima. An optimal algorithm for the case of two targets will also be presented. Finally in section 3.5 we will provide some concluding remarks.

## 3.1 Problem Definition

In the dynamic problem the time duration of the offense's attack is divided into a number of time segments. Each segment is of sufficient length to allow the defense to use a subset of its weapons and observe (perfectly) the outcomes of all of the engagements of the weapons. With the feedback of this information the defense can make better use of its weapons, since it will no longer engage targets which have already been destroyed. Thus we are dealing with so-called "shoot-look-shoot-..." strategies.

We assume that in the initial stage the defense chooses a subset of its weapons and assigns them to targets. These weapons are then committed simultaneously. In the second stage the outcomes (i.e. the survival or destruction of each engaged target) of all of the engagements of the weapons committed in the first stage are observed. Based on this observation, the defense chooses a subset of the remaining weapons and assigns them to the targets which survived the stage 1 engagements. In the third stage the outcomes of the engagements of the weapons committed in stage two are observed. Based on this observation, a subset of the remaining weapons is chosen and assigned to the set of surviving targets. This process is repeated for all time stages. In each stage the weapons are chosen and assigned with the objective of minimizing the total expected value of the surviving targets at the end of the final stage.

Note that in each stage the problem is resolved based on the outcomes of the previous stage. This implies that in each stage one is interested in obtaining (a) the subset of weapons which are to be fired in that stage and (b) the optimal assignment of these weapons to targets. Note that in computing the optimal assignment for the present stage one must assume that in all subsequent stages an optimal assignment will be used. If this is not done then the expected cost for the problem could be improved by doing so. This is known as the *Principle of Optimality* in dynamic programming [16]. We will therefore implicitly assume that optimal assignments will be used in all subsequent stages.

Note that the only information required to compute the optimal assignments in a stage is the set of surviving targets, the set of remaining weapons and the number of stages left. All other information of previous stages is not relevant. Therefore at each stage the problem can be restated as one in which the present stage is the *first* stage of the restated problem. The initial set of targets for this problem is the set of surviving targets and the initial set of weapons is the set of remaining weapons. In other words the problem to be solved in each stage has the same form as the statement of the problem for stage 1. Therefore, although we will only consider the $T$-stage problem and solve for the optimal assignments of the first stage, the same method can be used to solve for the optimal assignments of the remaining stages.

In our notation we will index the parameters in each stage with the stage number. Therefore for a $T$-stage problem the parameters in stage one will have an index of 1 while those of the final stage will have an index of $T$. The notation, which is basically the same as for the static problem except for the stage index, is as follows:

$N \stackrel{\text{def}}{=}$ the number of targets (offense weapons),

$M \stackrel{\text{def}}{=}$ the number of defense weapons,

$T \stackrel{\text{def}}{=}$ the number of time stages,

$V_i \stackrel{\text{def}}{=}$ the value of target $i$, $\quad i = 1, 2, \ldots, N$,

$p_{ij}(t) \stackrel{\text{def}}{=}$ the kill probability of weapon $j$ on target $i$ in stage $t$,

$i = 1, 2, \ldots, N, \qquad j = 1, 2, \ldots, M,$

$q_{ij}(t) \equiv \quad 1 - p_{ij}(t)$, the corresponding survival probability.

The decision variables will be denoted by:

$$x_{ij} = \begin{cases} 1 & \text{if weapon } j \text{ is assigned to target } i \text{ in stage 1} \\ 0 & \text{otherwise.} \end{cases}$$

The *target state* of the system in stage 2 will be defined as the set of targets which survive stage 1. This state will be denoted by an $N$-dimensional binary vector $\vec{u} \in \{0,1\}^N$ and represented by

$$u_i = \begin{cases} 1 & \text{if target } i \text{ survives stage 1,} \\ 0 & \text{if target } i \text{ is destroyed in stage 1.} \end{cases}$$

The *weapon state* of the system in stage 2 will be defined as the set of available weapons after stage 1. This state will be denoted by an $M$-dimensional binary vector $\vec{w} \in \{0,1\}^M$ and represented by

$$w_j = \begin{cases} 1 & \text{if weapon } j \text{ was not used in stage 1,} \\ 0 & \text{if weapon } j \text{ was used in stage 1.} \end{cases}$$

Given a first stage assignment, $\{x_{ij}\}$, the target state at the start of the second stage is an $N$-dimensional random vector. The probability that $u_i$ is 1 is the probability that target $i$ survives the first stage. The probability that $u_i$ is 0 is the probability that target $i$ is destroyed in the first stage. The distribution of the random variable $u_i$ is therefore given by:

$$\Pr[u_i = k] = k \prod_{j=1}^{M} (1 - p_{ij}(1))^{x_{ij}} + [1 - k] \left\{ 1 - \prod_{j=1}^{M} (1 - p_{ij}(1))^{x_{ij}} \right\}, \tag{3.1}$$
$$\text{for} \qquad k = 0, 1, \qquad i = 1, 2, \ldots, N.$$

Equation 3.1 will be called the *target state evolution* of the system.

The weapon state also evolves with time. This evolution is deterministic and depends on the assignments made in the first stage. The evolution is given by:

$$w_j = 1 - \sum_{i=1}^{N} x_{ij}, \qquad j = 1, 2, \ldots, M. \tag{3.2}$$

This simply says that weapon $j$ is available in the second stage if and only if it is not used in the first stage. Equation 3.2 will be called the *weapon state evolution* of the system.

We will let $F_2^*(\vec{u}, \vec{w})$ denote the *optimal* cost of a $T-1$ stage problem with initial target state $\vec{u}$ and initial weapon state $\vec{w}$. Note that this problem will be defined in terms of optimal costs for $T-2$-stage problems, etc. Eventually the $(T-(T-1)$ or single stage problem will be defined in terms of optimal costs for 0-stage problems. The optimal cost of a 0-stage problem will be defined as:

$$F_{T+1}^*(\vec{u}, \vec{w}) = \sum_{i=1}^{N} V_i u \tag{3.3}$$

In other words, the cost is simply the total value of the targets which survived the final stage.

---

**Problem 3.1** *The Dynamic Target-Based problem (DTB) can be stated as:*

$$\min_{\{x_{ij}\}} F_1 = \sum_{\vec{\omega} \in \{0,1\}^N} \Pr[\vec{u} = \vec{\omega}] F_2^*(\vec{\omega}, \vec{w})$$

$$\text{subject to} \quad x_{ij} \in \{0,1\}, \qquad i = 1, 2, \ldots, N \quad j = 1, 2, \ldots, M,$$

$$\text{with} \quad w_j = 1 - \sum_{i=1}^{N} x_{ij}.$$

---

The objective function is the sum over all possible stage 2 target states of the probability of occurrence of that state times the optimal cost given that state. The probability distribution of the target state was given in 3.1. Note that the distribution of the stage 2 target state and the stage 2 weapon state both depend on the first stage assignment. The first constraint restricts each weapon to be assigned at most once in the first stage. The second constraint is due to the weapon state evolution.

This problem is considerably more difficult than the static one. This can be illustrated by attempting to use a straightforward dynamic programming approach to the problem. Let us consider a two stage problem. The number of possible weapon subsets that can be chosen in the first stage is $2^M$. If $m_1$ weapons are used in stage 1 the number of possible assignments that must be checked is $N^{m_1}$. If $\bar{N}$ of the $N$ targets are engaged in the first stage the number of possible outcomes is $2^{\bar{N}}$. If $\tilde{N}$ of the $N$ targets survive stage 1 and $m_2$ weapons are available in stage 2 then the number of assignments that must be checked to obtain the optimal cost for this outcome is $\tilde{N}^{m_2}$. These numbers show the enormous number of computations that will be required if a straightforward dynamic programming approach is used. Note that to simply evaluate the expected value of a first stage assignment requires a tremendous computational effort.

## 3.2 Unit Valued Targets and Stage Dependent Kill Probabilities

In this section we will study the effect of stage dependent kill probabilities $p(t)$ on the optimal assignment. We will assume that the targets all have a value of unity and that the kill probabilities $p(t)$ are independent of the weapons and the targets. We were not able to obtain an analytical solution to this problem even for the case of two targets. However, we were able to obtain results for the limiting case, as the number of targets goes to infinity. We will first present some properties of the optimal solution.

**Theorem 3.1** *Consider the dynamic version of the Target-Based problem in which there are $T$ stages, $N$ unit-valued targets, stage dependent kill probabilities $p(t)$, and $M$ weapons. The optimal strategy has the property that the weapons to be used at each stage are spread as evenly as possible among the surviving targets.*

*Proof:* The proof is by contradiction. To simplify the notation we will denote the kill probability for the first stage by $p$ (instead of $p(1)$) and the survival probability by $q = 1 - p$. Similarly we will denote the optimal number of weapons assigned to target $i$ in the first stage by $x_i$. Let us assume that assignment $\bar{x}$ is optimal but does not have the property that the weapons are spread evenly among the targets. For convenience let us assume that targets 1 and 2 are such that $x_1 > x_2 + 1$. The remaining assignments $x_3, x_4, \ldots, x_N$ can be arbitrary. Denote the expected cost for this optimal assignment by $F^*$. We have

$$F^* = q^{x_1+x_2} D_2 + [q^{x_1}(1 - q^{x_2}) + q^{x_2}(1 - q^{x_1})] D_1 + (1 - q^{x_1})(1 - q^{x_2}) D_0 \qquad (3.4)$$

where

$D_0 \stackrel{\text{def}}{=}$ the expected cost given that targets 1 and 2 are destroyed in stage 1,

$D_1 \stackrel{\text{def}}{=}$ the expected cost given that either target 1 or target 2 survives stage 1,

$D_2 \stackrel{\text{def}}{=}$ the expected cost given that targets 1 and 2 both survive stage 1.

Now consider the assignment in which a single weapon is removed from target 1 and reassigned to target 2. Denote the expected cost for this assignment by $F$. We have

$$F = q^{x_1+x_2}D_2 + [q^{x_1-1}(1-q^{x_2+1}) + q^{x_2+1}(1-q^{x_1-1})]D_1 + (1-q^{x_1-1})(1-q^{x_2+1})D_0 \quad (3.5)$$

where $D_0, D_1$ and $D_2$ are as defined above. We therefore have

$$\begin{aligned} F^* - F &= (1-q)(q^{x_2} - q^{x_1-1})D_1 + (1-q)(q^{x_1-1} - q^{x_2})D_0 \\ &= (1-q)(q^{x_2} - q^{x_1-1})(D_1 - D_0). \end{aligned} \quad (3.6)$$

By our assumption that $x_1 > x_2 + 1$, we have $q^{x_2} > q^{x_1-1}$. Also, since $D_1$ is the expected cost given that either target 1 or target 2 survives and $D_0$ is the cost if both are destroyed then $D_1 > D_0$. This implies that the expression 3.6 is positive which implies that $F^* > F$ which contradicts our assumption that the assignment with expected cost $F^*$ was optimal. ■

The above result simplifies the problem to be solved since we can use the number of weapons to be used each stage, $m_t$, as the decision variable and optimize over this variable. Given the optimal values of $m_t$, the optimal assignment can be obtained by spreading these weapons evenly among the targets. In the case of $T = 2$ the resulting problem is a one dimensional optimization problem since $m_1 + m_2 = M$. Intutively we would expect the expected cost to be a unimodal[1] function with respect to the number of weapons used in stage 1. However, this is not the case as we see in the following two-stage example.

Let us choose $m_1$, the number of first stage weapons, as the independent variable. We will write the expected value if $m_1$ weapons are used in stage 1 and $M - m_1$ weapons are used in stage 2 by $F_1(m_1)$. The optimal solution can then be obtained by minimizing $F_1(m_1)$ over the set $\{0, 1, \ldots, M\}$.

[1]See definition A.5.

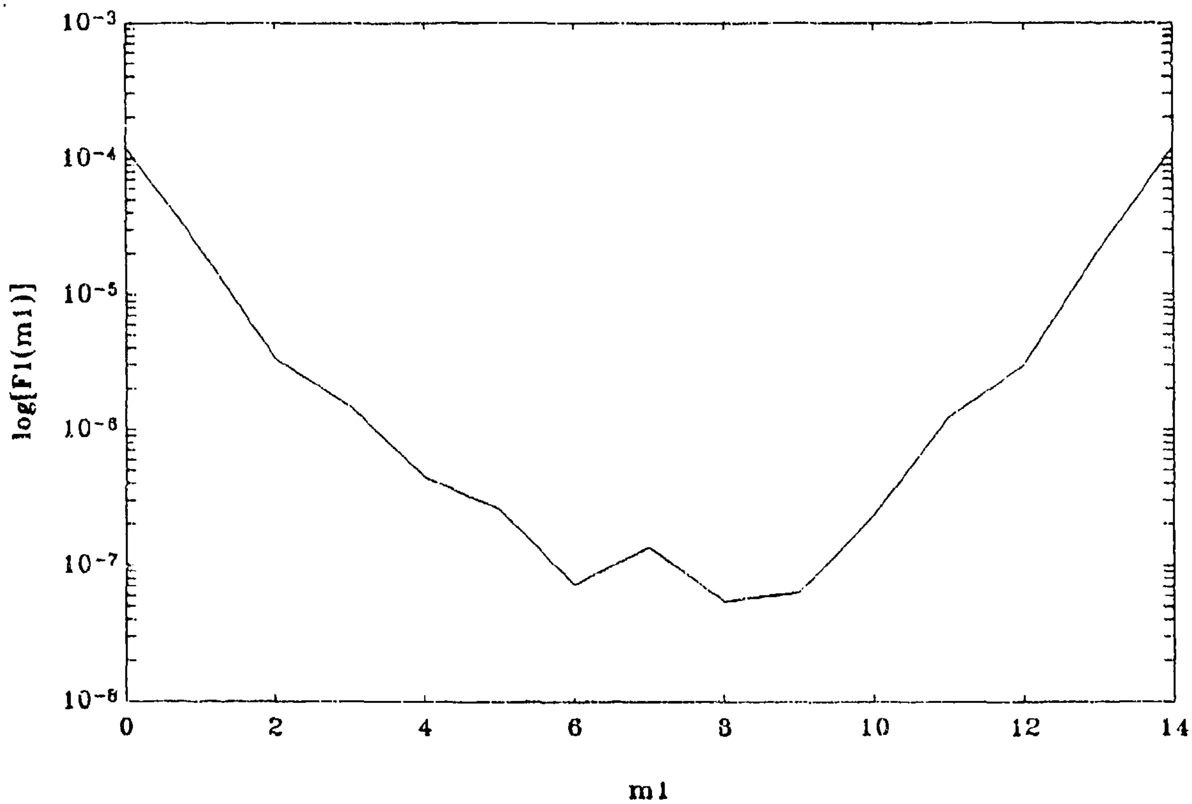


Figure 3.1: A two-stage example in which the expected cost as a function of the number of first stage weapons, $F_1(m_1)$, has multiple local minima.

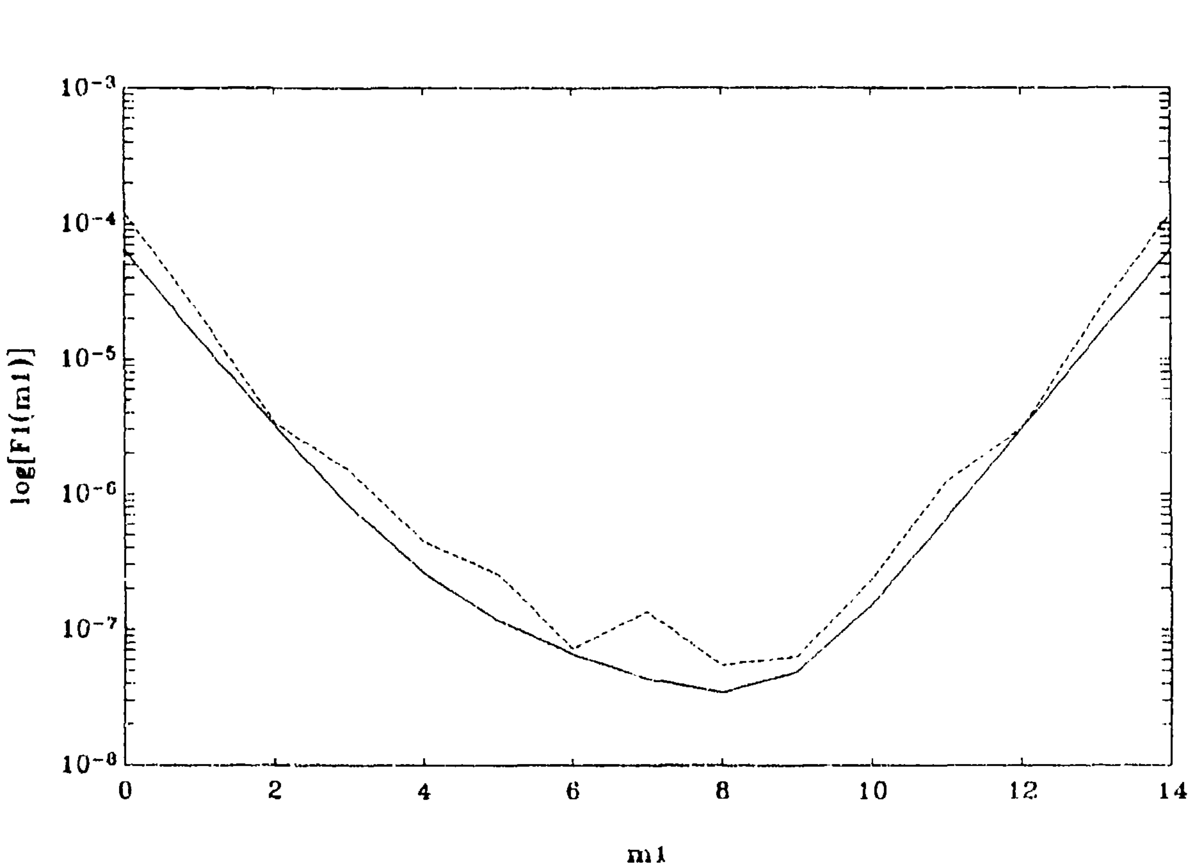


Figure 3.2: The log of the function $F_1(m_1)$ for the relaxed problem (solid) as well as for the true problem (dashed) vs. the number of first stage weapons, $m_1$.

If $F_1(m_1)$ was a unimodal function of $m_1$ then the above minimization could be done efficiently by using a local search algorithm. Unfortunately, this is not the case as can be seen in the following example. Consider the problem in which $T = 2$, $M = 14$, $N = 3$ and $p(1) = p(2) = 0.9$. In Figure 3.1 we have plotted $\log F_1(m_1)$ versus $m_1$. We used a log scale because the variations near the global minima are so small, that with a linear scale the function "appears" to have a single minimum. This suggests that for all practical purposes any of the local minima will suffice. A local minimum can easily be obtain d by a local search algorithm (i.e. repeatedly increase or decrease $m_1$ if doing so decreases the cost until any change in $m_1$ results in an increase in the cost.

The function $F_1(m_1)$ is not unimodal because of the restriction that weapons must be assigned in integral quantities. Let us consider the problem in which the integrality constraint is relaxed. In other words we will assume that the survival probability of each target in stage one is given by $(1-p)^{\frac{m_1}{N}}$. Also if $k$ targets survive stage one we will assume that the expected cost in stage two is given by $k(1-p)^{\frac{M-m_1}{k}}$. In figure 3.2 we have plotted the expected cost for this relaxed problem (the solid curve) versus the number of weapons used in stage one. We also plotted the function for the case of the true problem, $F_1(m_1)$, (the dashed curve). Note that the function of the relaxed problem is a lower bound for the function of the true problem. This is expected since the set of feasible solutions of the relaxed problem includes the set of feasible solutions of the true problem. Note that the function for the relaxed problem is unimodal. Also note that the optimal value of $m_1$ is the same for both functions. This suggests that the relaxed problem can be used to find a near-optimal value for $m_1$. There are two advantages to using the relaxed problem. First of all, it is easier to solve because of the relaxation of the integral constraints. Secondly, if the function is unimodal, as it is for this case, then the global optimal can be found with the use of a local search algorithm.

Our next theorem concerns the case in which the number of weapons is less than the number of targets. Our intuition tells us that a dynamic allocation should not perform any better than a static one. This is indeed the case.

**Theorem 3.2** *If $M \leq N$, then the optimal strategy is to assign all of the weapons in the stage with the highest kill probability.*

*Proof:* This is of course true for the one stage problem. Let us assume that it is true for the $T-1$ stage problem. Now consider the $T$-stage problem. Suppose we assign $m_1$ weapons in the first stage. We have $m_1 \leq M \leq N$. By the induction assumption, all of the remaining weapons will be assigned in one of the following stages, the stage with maximum kill probability. This means that the problem can be reduced to a two stage problem. We will therefore assume that $T = 2$. The cost for this problem is given by

$$\begin{aligned} F_1(m_1) &= \sum_{j=1}^{m_1} b(j; m_1, p(1))[(M - m_1)(1 - p(2)) + (j + N - M)] \\ &= (M - m_1)(1 - p(2)) + m_1(1 - p(1)) + N - M \\ &= N - Mp(2) + m_1[p(2) \quad p(1)]. \end{aligned}$$

Therefore if $p(2) > p(1)$, then $m_1^* = 0$; while if $p(2) < p(1)$, then $m_1^* = M$. In the case $p(1) = p(2)$, both solutions are optimal. Therefore for the $T$-stage problem all weapons are assigned in the stage with the maximum kill probability. ■

The above theorem is not particularly enlightening. However, it allows us to concentrate on the cases $M \geq N$. Our next result pertains to these cases. It states that, if $M \geq N$ and $p(t) \geq p(t+1)$ then the optimal assignment has the property $m_1^* \geq N$. In other words the optimal number of weapons to be used in the first stage is at least as big as the number of targets.

**Theorem 3.3** *If $M \geq N$, and $p(t) \geq p(t+1)$ for $t = 1, 2, \ldots, T-1$, then $m_1^* \geq N$.*

*Proof:* Note that the theorem is true for the case $T = 1$. Let us now assume that it is true for the $T-1$-stage problem and show that it holds for the $T$-stage problem. We will assume that the weapons to be used in a stage are spread evenly among the surviving targets.

The following notation will be used. Let $F_1(m_1)$ denote the expected cost given that $m_1$ weapons are used in stage 1 and that an optimal strategy is used for the remaining $T-1$ stages. Let $F_2(m, k)$ denote the optimal cost of the $T-1$ stage problem given that $m$ weapons are available and that $k$ targets survive stage 1. Assume that $m_1 = \bar{m} < N$. Note that

$$F_1(\bar{m}) = \sum_{k=0}^{\bar{m}} b(k; \bar{m}, q(1)) F_2(M - \bar{m}, N - \bar{m} + k)$$

By our assumptions, at least one target survives stage 1. Also at least one weapon is assigned in stage 2. Choose any target $i$ to which at least one weapon is assigned in stage

2. Let $x$ be the number of weapons assigned to this target. Denote one of these weapons as weapon $r$. We have:

$$\begin{aligned} F_2(M-\bar{m}, N-\bar{m}+k) &= q(2)(\text{Cost given that weapon } r \text{ hits target}) + \\ & \quad p(2)(\text{Cost given that weapon } r \text{ misses target}). \end{aligned}$$

The cost given that the weapon does not destroy the target is the same as the optimal cost for the case with $M-\bar{m}-1$ weapons. The cost given that the weapon destroys the target is the same as the optimal cost for the case of $N-\bar{m}+k-1$ targets and $M-\bar{m}-x$ weapons. Therefore

$$\begin{aligned} F_2(M-\bar{m}, N-\bar{m}+k) &= q(2)F_2(M-\bar{m}-1, N-\bar{m}+k) \\ & \quad +p(2)F_2(M-\bar{m}-x, N-\bar{m}+k-1). \end{aligned}$$

The fact that $x \geq 1$ can now be used to obtain

$$\begin{aligned} F_2(M-\bar{m}, N-\bar{m}+k) &\geq F_2(M-\bar{m}-1, N-\bar{m}+k) \\ & \quad +p(2)[F_2(M-\bar{m}-1, N-\bar{m}+k-1) \\ & \quad -F_2(M-\bar{m}-1, N-\bar{m}+k)] \end{aligned}$$

We now use the fact that $p(1) \geq p(2)$ to obtain

$$\begin{aligned} F_2(M-\bar{m}, N-\bar{m}+k) &\geq F_2(M-\bar{m}-1, N-\bar{m}+k) \\ & \quad +p(1)[F_2(M-\bar{m}-1, N-\bar{m}+k-1) \\ & \quad -F_2(M-\bar{m}-1, N-\bar{m}+k)] \end{aligned} \tag{3.7}$$

Next note that by using an argument similar to that above we can show that

$$\begin{aligned} F_1(\bar{m}+1) &= \sum_{k=0}^{\bar{m}} b(k;\bar{m},q(1))\{F_2(M-\bar{m}-1, N-\bar{m}+k) \\ & \quad +p(1)[F_2(M-\bar{m}-1, N-\bar{m}+k-1) - F_2(M-\bar{m}-1, N-\bar{m}+k)]\} \end{aligned} \tag{3.8}$$

We therefore have

$$F_1(\bar{m}) = \sum_{k=0}^{\bar{m}} b(k;\bar{m},q(1))F_2(M-\bar{m}, N-\bar{m}+k)$$

Substituting the inequality we obtained in 3.7 we have

$$\begin{aligned} F_1(\bar{m}) \geq & \sum_{k=0}^{\bar{m}} b(k;\bar{m},q(1))\{F_2(M-\bar{m}-1,N-\bar{m}+k) \\ & +p(1)[F_2(M-\bar{m}-1,N-\bar{m}+k-1)-F_2(M-\bar{m}-1,N-\bar{m}+k)]\} \end{aligned} \tag{3.9}$$

Now using the result obtained in 3.8 we have

$$F_1(\bar{m}) \geq F_1(\bar{m}+1)$$

This means that the expected cost can be decreased by increasing the number of weapons used in the first stage. This process can be repeated to conclude that the optimal number of weapons to be used in the first stage must be greater than or equal to the number of targets. ∎

The next theorem concerns the case in which the number of stages is large. One would expect that if this is the case then at each stage one should assign a single weapon to each surviving target at that stage. If two weapons are assigned to a target at a stage and one of them destroys the target then the other weapon has essentially been wasted. This result is given in the following theorem.

**Theorem 3.4** *If $T > 1 + \frac{M-N}{2}$, $M > N$ and $p(t) = p$ for $t = 1,\ldots,T$ then $m_1^* = N$.*

*Proof:* Assume that $m_1^* > N$. This means that there exists at least one target $i$ with $x_i^* > 1$. Suppose that we increase the number of stages by one. This additional stage will be added at the beginning and we will assign $m_1^*$ weapons over two stages (instead of one). This assignment will be as follows. In stage 1 a single weapon per target will be assigned. In the second stage $x_i^* - 1$ weapons will be assigned to target $i$ *whether or not it is destroyed in stage one* (i.e use an open loop strategy). Note that the expected cost of this strategy is the same as for the orginal $T$-stage problem. Therefore, if we allow the weapons in stage 2 to be assigned to any of the surviving targets then the expected cost can only decrease. In other words, by increasing the number of stages by one we can decrease the cost by using at most

| | $p(2)$=0.1 | $p(2)$=0.3 | $p(2)$=0.5 | $p(2)$=0.7 | $p(2)$=0.9 |
|---|---|---|---|---|---|
| $p(1) = 0.1$ | 10 | 0 | 0 | 0 | 0 |
| $p(1) = 0.3$ | 20 | 10 | 0 | 0 | 0 |
| $p(1) = 0.5$ | 20 | 16 | 10 | 10 | 0 |
| $p(1) = 0.7$ | 20 | 18 | 10 | 10 | 10 |
| $p(1) = 0.9$ | 20 | 20 | 10 | 10 | 10 |

Table 3.1: Optimal values of first-stage weapons, $m_1$, for different combinations of stage dependent kill probabilities.

| | $p(2)$=0.1 | $p(2)$=0.3 | $p(2)$=0.5 | $p(2)$=0.7 | $p(2)$=0.9 |
|---|---|---|---|---|---|
| $p(1) = 0.1$ | 8.0100 | 4.9000 | 2.5000 | 0.9000 | 0.1000 |
| $p(1) = 0.3$ | 4.9000 | 4.2784 | 2.5000 | 0.9000 | 0.1000 |
| $p(1) = 0.5$ | 2.5000 | 2.3975 | 1.4334 | 0.6496 | 0.1000 |
| $p(1) = 0.7$ | 0.9000 | 0.8924 | 0.4808 | 0.1626 | 0.0238 |
| $p(1) = 0.9$ | 0.1000 | 0.1000 | 0.0410 | 0.0079 | 0.0005 |

Table 3.2: Optimal costs for various combinations of stage dependent kill probabilities.

$N$ weapons in the first stage. Therefore the optimal solution must have this property. From theorem 3.3 we know that at each stage at least one weapon should be assigned to each target. These two results imply that in the optimal solution exactly one weapon should be assigned to each target in stage one.

Note that the maximum possible number of stages is obtained in the case in which 2 targets survive stage one and they survive all future stages. In this case the number of stages in which weapons are available is given by $1 + \frac{M-N}{2}$. Hence if $T$ is greater than this number the problem can be considered as having an infinite number of stages. ∎

### 3.2.1 Numerical Results

Since there appears to be no analytical solution to the problem under the assumption of unit-valued targets and stage dependent kill probabilities, we numerically computed optimal solutions for a simple example. We computed the optimal solutions for the case of $M = 20$ weapons, $N = 10$ unit-valued targets and $T = 2$ stages for various kill probabilities. If we denote the kill probability for the first stage by $p(1)$ and that for the second stage by $p(2)$, then our solutions are for the cases in which $p(1) = .1, .3, .5, .7, .9$, $p(2) = .1, .3, .5, .7, .9$.

Table 3.1 contains the optimal numbers of weapons to be used in the first stage (i.e. $m_1$). Note that when $p(2)/p(1)$ is large then $m_1^*$ is small while when $p(2)/p(1)$ is small $m_1^*$ is large. In certain ranges we find that $m_1$ is very sensitive to $p(2)$ and $p(1)$. For example, when the second-stage kill probability is $p(2) = 0.9$, then a change in the first-stage kill probability $p(1)$ from 0.5 to 0.7 results in the optimal number of first-stage weapons, $m_1^*$, changing from 0 to 10 which means a change from a static to a dynamic strategy.

Table 3.2 contains the optimal costs for the problems. Note that, given a choice between using the more effective weapons in the first or second stage, it is always better to use them in the first stage. For example, if $p(2) = 0.9$ and $p(1) = 0.7$, the optimal cost is 0.0238; while if $p(2) = 0.7$ and $p(1) = 0.9$, the optimal cost is 0.0079. Another way of looking at this property is the following. We computed (approximately) the derivative of the optimal cost with respect to $p(2)$ at the point $p(2) = 0.8, p(1) = 0.8$ to be -0.22. The derivative with respect to $p(1)$ at the same point is approximately -0.33. This implies that, given the choice of improving the kill probabilities of either the weapons used in the first stage or the weapons used in the second stage, one should improve the kill probabilities of the weapons for the first stage.

### 3.2.2 The Limit of an Infinite Number of Targets

In this section we will consider what happens for very large numbers of unit-valued targets, $N$. We will keep the ratio of weapons to targets fixed and solve the problem in the limit as the number of targets goes to infinity. We will find that, in the limit, the problem can be considered as a deterministic one in which the number of targets in a stage is the expected number of targets which survive the previous stage.

Let us introduce the variable $\kappa_t \equiv \frac{m_t}{N}$. This is the number of weapons reserved for stage $t$ per initial number of targets. We will also define the *vector* $\vec{\kappa_t} \in \Re^t$ for $1 \leq t \leq T$ by

$$\vec{\kappa}_t \equiv [\kappa_t, \kappa_{t+1}, \ldots, \kappa_T]^T.$$

Note that the values of $\kappa_t$ may not be optimal for the problem. We will address the question of finding optimal values for $\kappa_t$ in subsection 3.2.3. By theorem 3.1 we know that the weapons to be used in each stage should be spread evenly among the surviving targets. The expected cost of the $T$-stage problem with $N$ targets and in which $m_t = \kappa_t N$ weapons are used in stage $t$ will be denoted by $F_1(N, \vec{\kappa}_1)$. Let $\alpha$ denote the expected fraction, of the initial number of targets, which survive stage 1 i.e.

$$\alpha = [1 - (\kappa_1 - \lfloor \kappa_1 \rfloor)p(1)](1 - p(1))^{\lfloor \kappa_1 \rfloor}. \tag{3.10}$$

Note that $\alpha$ is independent of $N$. Consider the case of the static problem (i.e. $T = 1$). We have

$$\frac{F_1(N, \kappa_1)}{N} \equiv \alpha = [1 - (\kappa_1 - \lfloor \kappa_1 \rfloor)p(1)](1 - p(1))^{\lfloor \kappa_1 \rfloor}.$$

Taking the limit as $N$ goes to infinity on both sides we get

$$\lim_{N \to \infty} \frac{F_1(N, \kappa_1)}{N} = [1 - (\kappa_1 - \lfloor \kappa_1 \rfloor)p(2)](1 - p(2))^{\lfloor \kappa_1 \rfloor} = \alpha. \tag{3.11}$$

In other words, for the static problem, if the weapon to target ratio is kept fixed then the expected fraction of targets which survive is the same for all values of $N$. This will also be the value in the limit as the number of targets goes to infinity. We will now show how the limit of this ratio can be obtained for more than a single stage. The limit for the $T$-stage problem will be obtained in terms of the limit for the $T - 1$ stage problem, etc. Since the limit for the case $T = 1$ (the static problem) is well defined then the limit for the two-stage problem is well defined etc. The $T$-stage limit is therefore well defined. The main result will now be presented.

**Theorem 3.5** *Consider the $T$-stage problem with $N$ unit valued targets, $M = \kappa N$ weapons and stage dependent kill probabilities $p(t)$. Assume that the number of weapons to be used in stage $t$ is given by $m_t = \kappa_t N$, where $\kappa_t \in [0, \kappa]$ is a fixed constant which may be different for each stage. We then have that*

$$\lim_{N \to \infty} \frac{F_1(N, \vec{\kappa}_1)}{N} = \alpha \lim_{N \to \infty} \frac{F_2(N, \vec{\kappa}_2/\alpha)}{N}, \tag{3.12}$$

*where $\alpha$ is given by equation 3.10.*

*Proof:* Let $N_2$ represent the number of targets which survive stage 1. $N_2$ is a random variable. If $\kappa_1 \in \aleph$ then it is a binomial random variable; otherwise, its distribution can be obtained by the convolution of two binomial distributions. The mean and variance of this distribution is given by:

$$E[N_2] \equiv \bar{N}_2 = \alpha N,$$

$$Var[N_2] \equiv \sigma^2 = \beta N,$$

where

$$\beta = q(1)^{\lfloor\kappa_1\rfloor}[(1-(\kappa_1-\lfloor\kappa_1\rfloor))(1-q(1)^{\lfloor\kappa_1\rfloor})+(\kappa_1-\lfloor\kappa_1\rfloor)q(1)(1-q(1)^{\lfloor\kappa_1\rfloor+1})]$$

Note that $\beta$ is independent of $N$. For any $\mu > 0$ we have

$$\begin{aligned} F_1(N,\vec{\kappa_1}) &= \Pr(|N_2-\bar{N}_2| \geq \mu N)E[F_2(N_2,\vec{\kappa}_2)||N_2-\bar{N}_2| \geq \mu N)] \\ &\quad + \Pr(|N_2-\bar{N}_2| < \mu N)E[F_2(N_2,\vec{\kappa}_2)||N_2-\bar{N}_2| < \mu N)]. \end{aligned} \tag{3.13}$$

By Chebyshev's Inequality we know that

$$\Pr(|N_2-\bar{N}_2| < \mu N) \geq 1 - \frac{\sigma^2}{(\mu N)^2} = 1 - \frac{\beta}{\mu^2 N}. \tag{3.14}$$

Since $F_2(N_2,\vec{\kappa}_2)$ is a monotonically increasing function of $N_2$ then

$$E[F_2(N_2,\vec{\kappa}_2)||N_2-\bar{N}_2| < \mu N] \leq F_2(\bar{N}_2+\mu N,\vec{\kappa}_2), \tag{3.15}$$

and

$$E[F_2(N_2,\vec{\kappa}_2)||N_2-\bar{N}_2| < \mu N] \geq F_2(\bar{N}_2-\mu N,\vec{\kappa}_2), \tag{3.16}$$

and also

$$E[F_2(N_2,\vec{\kappa}_2)||N_2-\bar{N}_2| \geq \mu N] \leq F_2(N,\vec{\kappa}_2), \tag{3.17}$$

Using 3.14 3.15, 3.16 and 3.17 in 3.13 we obtain

$$(1-\frac{\beta}{\mu^2 N})F_2(\bar{N}_2-\mu N,\vec{\kappa}_2) \leq F_1(N,\vec{\kappa_1}) \leq \frac{\beta}{\mu^2 N}[F_2(N,\vec{\kappa}_2)] + F_2(\bar{N}_2+\mu N,\vec{\kappa}_2) \tag{3.18}$$

Dividing by $N$ and taking the limit as $N$ goes to infinity we obtain

$$\lim_{N\to\infty}\frac{F_2(\bar{N}_2-\mu N,\vec{\kappa}_2)}{N} \leq \lim_{N\to\infty}\frac{F_1(N,\vec{\kappa_1})}{N} \leq \lim_{N\to\infty}\frac{F_2(\bar{N}_2+\mu N,\vec{\kappa}_2)}{N}.$$

Using the fact that $\tilde{N}_2 = \alpha N$ and taking $\mu$ arbitrarily close to 0 we obtain

$$\lim_{N\to\infty} \frac{F_1(N,\vec{\kappa}_1)}{N} = \lim_{N\to\infty} \frac{F_2(\alpha N,\vec{\kappa}_2)}{N}.$$

Using a change of variables we finally obtain

$$\lim_{N\to\infty} \frac{F_1(N,\vec{\kappa}_1)}{N} = \alpha \lim_{N\to\infty} \frac{F_2(N,\vec{\kappa}_2/\alpha)}{N}.$$

This completes the proof. ∎

Note that the theorem gives the limit of the $T$-stage problem in terms of the limit for a $(T-1)$-stage problem. The latter can be expressed in terms of the limit of a $(T-2)$-stage problem etc. The limit for the case $T = 1$ is given in equation 3.11.

This limit provides us with a lower bound for finite values of $N$. This result is given in the next theorem.

**Theorem 3.6** *Consider the $T$-stage problem with $N$ unit valued targets, $M = \kappa N$ weapons and stage dependent kill probabilities $p(t)$. Assume that the number of weapons to be used in stage $t$ is given by $m_t = \kappa_t N$, where $\kappa_t \in [0, \kappa]$ is a fixed constant which may be different for each stage. We then have that*

$$F_1(N,\vec{\kappa}_1) \geq N \lim_{\hat{N}\to\infty} \frac{F_1(\hat{N},\vec{\kappa}_1)}{\hat{N}}. \tag{3.19}$$

*Proof:* Let $k \in \aleph$, be any positive integer. Consider the problem with $kN$ targets and in which $m_t = k\kappa_t N$ weapons are used in stage $t$. Let $F_1(kN,\vec{\kappa}_1)$ denote the optimal cost for this problem. A sub-optimal solution for this problem is the following. Split the problem into $k$ subproblems. Each of these subproblems has $N$ targets and uses $m_t = \kappa_t N$ weapons in each stage. The optimal cost for the problem under this restriction is given by $kF_1(N,\vec{\kappa}_1)$. Since this solution is suboptimal we have:

$$F_1(kN,\vec{\kappa}_1) \leq kF_1(N,\vec{\kappa}_1).$$

Dividing both sides by $kN$ and taking the limit as $k$ goes to infinity we have

$$\frac{F_1(N,\vec{\kappa}_1)}{N} \geq \lim_{k\to\infty} \frac{F_1(kN,\vec{\kappa}_1)}{kN} = \lim_{\hat{N}\to\infty} \frac{F_1(\hat{N},\vec{\kappa}_1)}{\hat{N}}.$$

The result 3.19 now follows. ∎

Theorem 3.6 provides us with a lower bound on the optimal cost for the problem with finite values of $N$. Theorem 3.5 is more easily understood if we look at some examples.

Example 1

Suppose that $\kappa = 2, \kappa_1 = 0.5, \kappa_2 = 1.5$ and $p = 0.6$. In other words the defense has $2N$ weapons, $N/2$ weapons are used in stage 1 and the remainder are used in stage 2. The expected fraction of targets which survive stage 1 is given by

$$\alpha = \frac{1}{2}[(1-p)+1] = 0.7.$$

Therefore the expected value in stage 2 given that the expected number of targets survive stage 1 is given by:

$$F_2(\alpha,\kappa_2) = F_2(0.7, 1.5) = [(1-p)^3 + 6(1-p)^2]/10 = 0.1024$$

Note that we had to scale the number of weapons and the number of targets by a factor of 10 so that there are an integral number of each. If we now use the theorem we obtain:

$$\lim_{N\to\infty} \frac{F_1(N,[.5,1.5])}{N} = 0.1024.$$

In words this says the following. For very large $N$, if 25% of the weapons are used in stage 1 then approximately 10% of the targets will survive both stages. For comparison, if a static strategy is used then 16% of the targets will survive. If we consider the case of two targets, $N = 2$, then 13.12% of the targets will survive both stages. Note that even for the case of $N = 2$ the limiting value provides a good approximation. This approximation gets better as $N$ increases.

Example 2

Suppose that $\kappa = 2, \kappa_1 = 1, \kappa_2 = 1$ and $p = 1 - \frac{1}{k}$ for some $k \in \aleph$. In this case $\alpha = (1-p)$ so that

$$F_2(\alpha, \kappa_2) = F_2((1-p), 1) = \frac{(1-p)^k}{k}.$$

Therefore

$$\lim_{N \to \infty} \frac{F_1(N, [1,1])}{N} = \frac{(1-p)^k}{k} = (1-p)^{\frac{2-p}{1-p}}.$$

Let us now consider the case of a three-stage problem.

Example 3

Consider the 3 stage problem with $\kappa = 3, \kappa_1 = \kappa_2 = \kappa_3 = 1$, and $p = 0.5$. In the limit the expected fraction of targets which survive stage 1 is $\frac{1}{2}$. The expected fraction which survive stage 2 is $\frac{1}{8}$ and the expected fraction which survives stage 3 is $\frac{1}{2^{11}}$. Therefore

$$\lim_{N \to \infty} \frac{F_1(N, [1,1,1])}{N} = 2^{-11}.$$

Let us now consider a case with stage dependent kill probabilities.

Example 4

Suppose that $\vec{\kappa_1} = [1.5, 1, .5]$ and that $p(1) = .6, p(2) = .5, p(3) = .4$. The expected fraction of targets which survive stage 1 is given by

$$\alpha = 0.5[(1-p(1)) + (1-p(1))^2] = 0.28.$$

The expected fraction which survives stage 2 is the solution to a static problem with 0.28 targets and 1 weapon. To find the limit for this problem we find the cost for the case of 7 targets and 25 weapons (i.e multiply by 2 ) and divide the cost by 25. We obtain

$$\alpha = [4(1-p(1))^4) + 3(1-p(1))^3]/25 = 0.025.$$

The expected fraction which survives the final stage is the solution to a static problem with 0.025 targets and .5 weapons. Multiplying the parameters by 40 etc. we obtain

$$\alpha = (1-p(2))^{20}/40 = 9.1 \times 10^{-7}.$$

Therefore in the limit as the number of targets goes to infinity, the expected fraction of the initial number of targets which survives all stages is $9.1 \times 10^{-7}$.

Theorem 3.5 is important because it allows us to compute approximate costs for the case of large $N$. We will show that this approximation is very good if $N > 100$. Theorem 3.6 says that this limit provides a lower bound on the cost for finite values of $N$.

In words theorem 3.5 says the following. Let us suppose that the number of weapons reserved for a stage is linearly dependent on the initial number of targets $N$. Therefore, as we increase the number of targets, the number of weapons in each stage will increase at the same rate. As we increase the number of targets, the expected number of targets which survive the final stage will also increase. Let us instead consider the ratio of the expected number of surviving targets and the initial number of targets. We will refer to this as the *expected surviving fraction of each target* since we can obtain the expected number of surviving targets by multiplying this ratio by $N$. We can compute this ratio in the limit of an infinite number of targets $N$ by solving a related deterministic problem. This deterministic problem is obtained as follows. Let us suppose that at each stage the number of surviving targets is equal to the *expected* number of surviving targets. Pick the initial number of target $N$ so that the exepcted number of surviving targets at each stage is integral. Using this value of $N$ we evaluate the expected surviving number of targets at the end of the final stage of the deterministic problem in which, at each stage the expected number of surviving targets survive the previous stage. The ratio of the expected number of surviving targets for this problem and the initial number of targets $N$ is the same as the ratio, in the limit as $N$ goes to infinity, of the expected number of surviving targets and the initial number of targets. Note that the former ratio is obtained by solving a deterministic problem while the latter ratio must be obtained by solving a *stochastic* problem for an infinite number of targets. This limit provides a lower bound for the ratio for finite values of $N$. Furthermore, it provides an approximate answer for large values of $N$. An interesting question is how

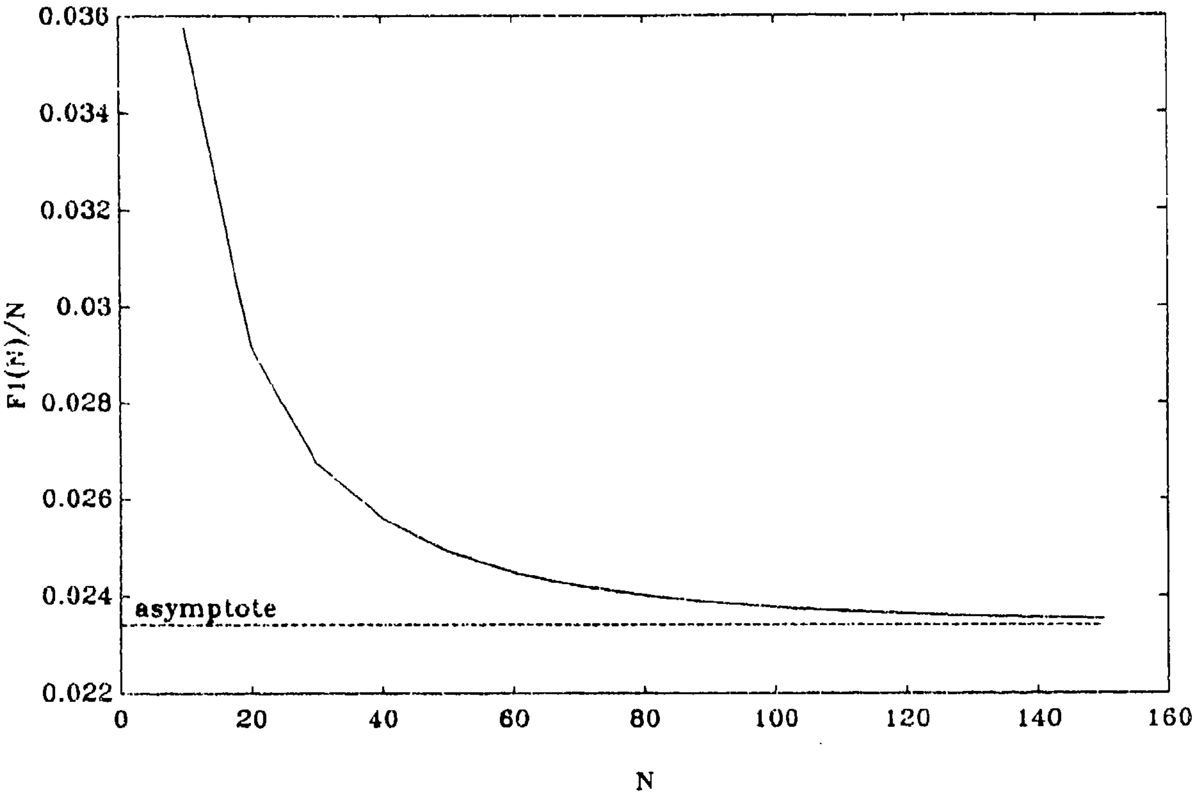


Figure 3.3: The ratio of the expected two-stage cost and the initial number of targets $N$ vs. $N$ for $p(1) = 0.6, p(2) = 0.7$; $N$ weapons are used in each stage.

large does $N$ have to be for the approximation to be good.

In order to address this question we have done the following. We have considered the problem of two stages $T = 2$ with $M = 2N$ weapons. $N$ weapons are used in each of the stages (i.e. $\bar{\kappa} = [1, 1]$). We computed the exact value of the ratio $\frac{F_1(N,\bar{x}_1)}{N}$ for $N = 10, 20, \ldots, 150$, and also in the limit as $N$ goes to infinity. In figure 3.3 we have plotted this ratio for finite values of $N$ as well as the ratio in the limit of infinite $N$. In this case we used a second stage kill probability of $p(2) = 0.7$ and a first stage kill probability of $p(1) = 0.6$. Figure 3.4 contains plots of the ratios for the case of $p(1) = p(2) = 0.5$. Figure 3.5 contains plots of the log of the ratios for the case of $p(1) = p(2) = 0.7$. Note that in each

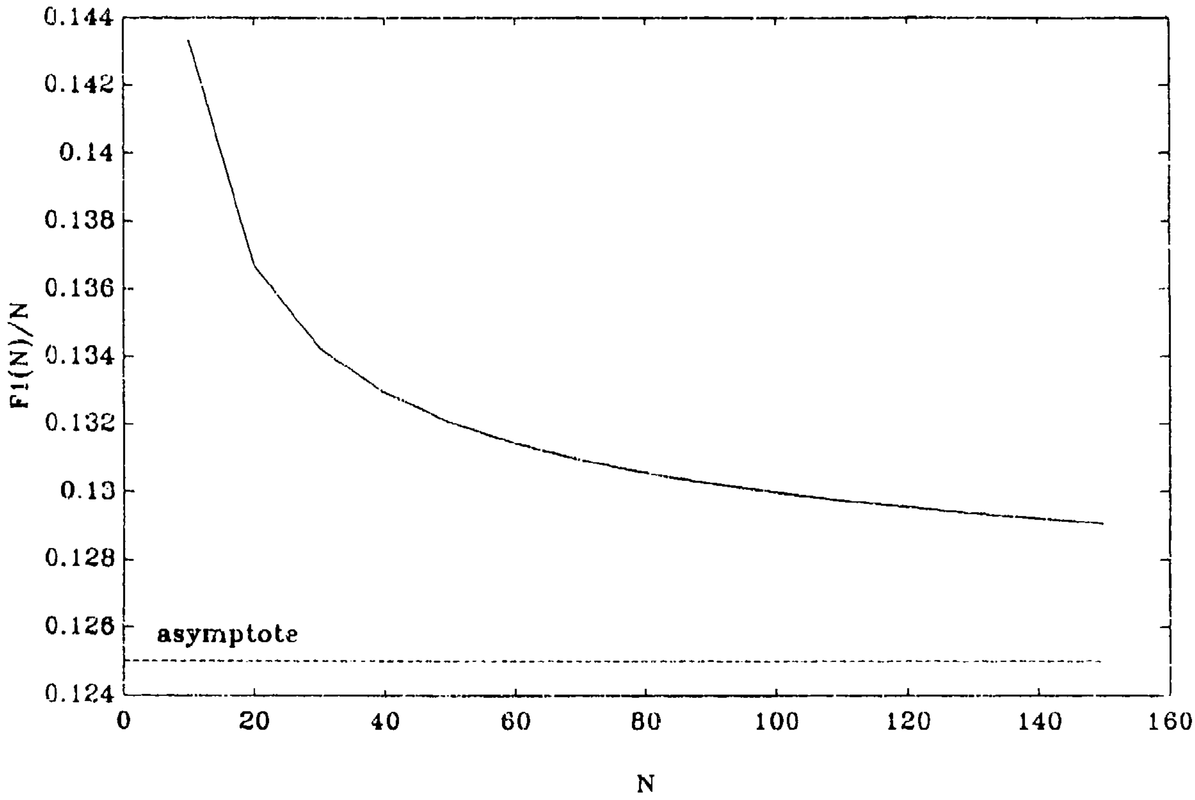


Figure 3.4: The ratio of the expected two-stage cost and the initial number of targets $N$ vs. $N$ for $p(1) = p(2) = 0.5$; $N$ weapons are used in each stage.

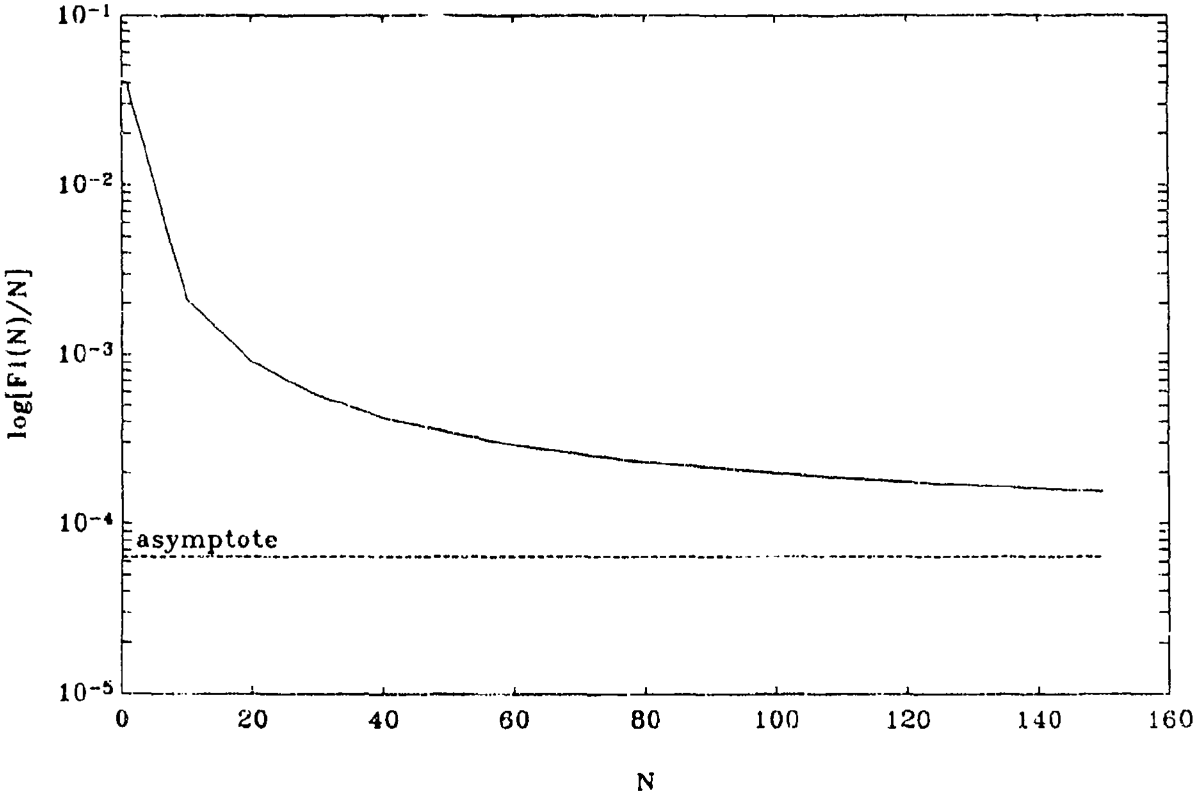


Figure 3.5: The log of the ratio of the expected two-stage cost and the initial number of targets $N$ vs. $N$ for $p(1) = p(2) = 0.7$; $N$ weapons are used in each stage.

of the cases the difference between the limiting value and the value for finite $N$ is small for $N > 100$.

### 3.2.3 Optimal Number of First-Stage Weapons for a Two-Stage problem with a Large Number of Targets

Note that in the discussion in the previous subsection the number of weapons to be used in each stage was fixed. In this section we will find optimal values for $\vec{\kappa}$ as $N$ goes to infinity. This will give us a good approximation to the optimal solution for large values of $N$.

We will only consider the two-stage case, $T = 2$. The optimization could also be attempted for $T > 2$, but it is doubtful whether one can find an analytical solution for such cases. For the case $T = 2$ we know that $\kappa_2 = \kappa - \kappa_1$ since all remaining weapons are used in the second stage. We therefore have a one dimensional optimization problem. We will let $\kappa_1$ be the free variable. The optimization problem can be stated as:

$$\min_{\kappa_1} F_2(\alpha, \kappa - \kappa_1) \tag{3.20}$$

$$\text{subject to} \qquad \kappa_1 \in [0, \kappa]$$

where

$$\alpha = [1 - (\kappa_1 - \lfloor \kappa_1 \rfloor) p(1)](1 - p(1))^{\lfloor \kappa_1 \rfloor}.$$

The function $F_2(\alpha, \kappa_2)$ is given by:

$$F_2(\alpha, \kappa_2) = [\alpha - p(2)(\kappa_2 - \alpha \lfloor \frac{\kappa_2}{\alpha} \rfloor) q(2)^{\lfloor \frac{\kappa_2}{\alpha} \rfloor}.$$

This expression is difficult to optimize. However, if the integrality constraint is relaxed, then the expected cost is given by $\alpha q(2)^{\frac{\kappa_2}{\alpha}}$. Since this is a lower bound for the non-relaxed problem, then

$$F_2(\alpha, \kappa_2) \geq \alpha q(2)^{\frac{\kappa_2}{\alpha}}. \tag{3.21}$$

This states that the solution obtained by allowing fractional assignments in the second stage is a lower bound to the solution in which only integral assignments are allowed. Note that if $\frac{\kappa_2}{\alpha} \in Z_+$ then equality holds in expression 3.21. Therefore, if the solution to the problem using the lower bound as the objective function is a multiple of $\alpha$ then it is optimal for the true problem.

The optimization problem using the lower bound in 3.21 as the objective function can be stated as:

$$\min_{\kappa_1} \alpha q(2)^{\frac{\kappa-\kappa_1}{\alpha}}. \tag{3.22}$$

$$\text{subject to} \qquad \kappa_1 \in [0, \kappa]$$

where

$$\alpha = [1 - (\kappa_1 - \lfloor \kappa_1 \rfloor)p(1)](1 - p(1))^{\lfloor \kappa_1 \rfloor}.$$

Let us first consider the case $\kappa = 1$. Note that theorem 3.2 has already provided us with a solution for this case. The solution is simply that all weapons should be assigned in the stage with the higher kill probability. Therefore,

$$\kappa_1^* = 0 \quad \text{for} \quad p(1) < p(2) \tag{3.23}$$

$$\kappa_1^* = 1 \quad \text{for} \quad p(1) \geq p(2) \tag{3.24}$$

Let us now consider the case in which $\kappa = 2$, i.e. a 2:1 weapon to target ratio. Using straightforward calculus one can show that the optimal values of $\kappa_1$ are given by

$$\kappa_1^* = 0 \quad \text{for} \quad \frac{2p(1) - 1}{p(1)} \leq \frac{1}{\log(1 - p(2))} \tag{3.25}$$

$$\kappa_1^* = 1 \quad \text{for} \quad \frac{2p(1) - 1}{p(1)[1 - p(1)]} \geq \frac{1}{\log(1 - p(2))} \geq \frac{-1}{p(1)} \tag{3.26}$$

$$\kappa_1^* = 2 \quad \text{for} \quad \frac{-1}{p(1)[1 - p(1)]} \geq \frac{1}{\log(1 - p(2))} \tag{3.27}$$

Note that if $\frac{1}{1-p(1)} \in \aleph$ then equality holds in 3.21. If this is the case then $\kappa_1^*$ is optimal for problem 3.20. Otherwise $\kappa_1^*$ is approximately optimal.

In the plot in figure 3.6 the vertical axis represents the kill probability in stage 1 while the horizontal axis represents the kill probability in stage 2. In each region we have indicated the optimal value of $m_1$, the number of weapons allocated in the first stage (recall that $m_1^* = \kappa_1^* N$) for the kill probabilities in that region. For example, consider the case $p(1) = 0.8$. If $0 < p(2) < 0.15$ then it is optimal to use all weapons in stage 1. If $0.15 < p(2) < 0.55$ then

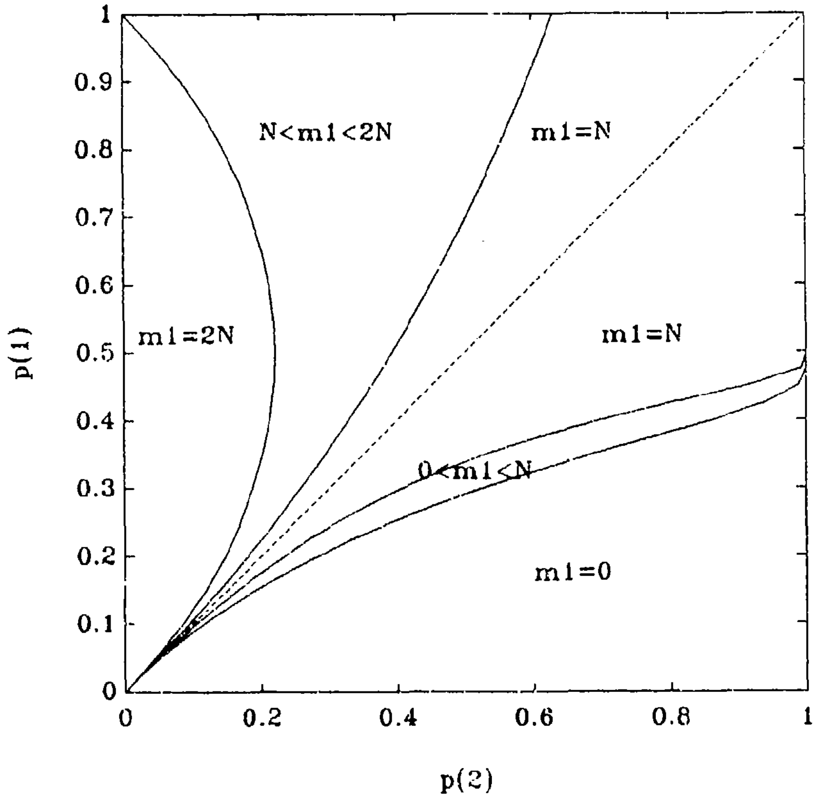


Figure 3.6: Optimal number of first-stage weapons, $m_1$, for various kill probabilities with $M = 2N$ weapons, in the limit of an infinite number of targets, $N$.

the optimal number of weapons to be used in stage 1 lies between $N$ and $2N$. If $p(2) > 0.55$ it is optimal to use half of the weapons in stage 1.

Note that for $0.6 \leq p(1) \leq 0.9$ and $0.6 \leq p(2) \leq 0.9$ it is optimal to use half of the weapons in stage 1. This implies that for the problems of interest to us (i.e large-scale problems with kill probabilities greater than 0.6) it is optimal to use half of the weapons in stage 1, even if the kill probabilities are different in each stage. This insensitivity of the optimal strategy to the kill probbabilities is very interesting. We should stress that this result is valid for large numbers of unit-valued targets and weapons

## 3.3 Unit Valued Targets and a Uniform Kill Probability

In the previous section we considered the case of stage dependent (but weapon and target independent) kill probabilities. In this section we will, in addition, assume that the kill probabilities are stage independent. We will assume that all targets have the same value and that the kill probability is the same for all weapon-target pairs in all stages. Without loss of generality we can assume that all targets have a value of unity. Under these assumptions we have shown (theorem 3.1) that the optimal strategy has the property that the weapons to be used at each stage must be spread evenly among the surviving targets, (Note that if the targets had unequal values then one would need to find the individual target assignments). Therefore the only decision variable is the number of weapons to be used in the present stage. Given this number, the optimal assignment for the present stage is obtained by spreading these weapons as evenly as possible among the surviving targets. The weapons are fired and the process is repeated. In this section we will present the optimal solution to this problem for the case of two targets. We will also present the solution in the limit as the number of targets goes to infinity (while keeping the weapon to target ratio fixed). These results will be used to compare dynamic and static strategies.

### 3.3.1 The Case of Two Targets

In this subsection we will make the additional assumption that $N = 2$. Although this is a simple problem, it has provided us with valuable insight. In the following theorem, we will show that, in the optimal strategy, the weapons are spread as evenly as possible among the stages left. Note that this theorem also holds for the case in which the targets have *different* values. Once the number of weapons to be used in the first stage is known the optimal assignment of these weapons to targets must be computed. We will see that this assignment can be obtained by solving a static problem.

**Theorem 3.7** *An optimal strategy for the special case of the Dynamic Target-Based problem in which $N = 2$ and $p_i(t) = p$ is as follows. Let $\bar{x}_1$ and $\bar{x}_2$ denote the optimal assignment of the two-target* **static** *problem with the same target values and kill probabilities as the dynamic problem but with $\lfloor \frac{M}{T} \rfloor$ weapons. The optimal decision variables for the dynamic problem is given by $x_1^* = \bar{x}_1, x_2^* = \bar{x}_2, m_1^* = \lfloor \frac{M}{T} \rfloor$.*

**Proof:** We will first prove the following lemma.

**Lemma 3.1** *If the number of weapons to be used in the present stage is fixed then the optimal assignment of those weapons is the same as the optimal assignment of the corresponding static problem with the same number of weapons.*

*Proof of Lemma:* Let us define $q$ by $q \equiv 1 - p$. Denote the optimal cost of the $t$-stage problem with $M$ available weapons by $F^*_{T-t+1}(M)$. Therefore the optimal cost for the $T$ stage problem will be denoted by $F^*_1(M)$. Let $x_1$ denote the number of weapons assigned to target 1 in stage 1 and let $x_2$ denote the number of weapons assigned to target 2 in stage 1. Note that if one of the targets is destroyed in the first stage, then it is optimal to assign all of the remaining weapons to the other target in the second stage. Therefore, the cost $F^*_1(M)$ can be obtained recursively by solving the following problem:

$$\begin{aligned} F^*_1(M) = \min_{\{x_1,x_2\}} \quad & \{q^{x_1}(1-q^{x_2})V_1 + q^{x_2}(1-q^{x_1})V_2\}q^{M-x_1-x_2} \\ & + q^{x_1+x_2}F^*_2(M - x_1 - x_2), \end{aligned} \tag{3.28}$$

$$\text{subject to} \qquad x_1 + x_2 \leq M, \qquad x_i \in Z_+.$$

If the number of weapons to be used in the first stage is restricted to $\bar{m}_1$, then the optimal cost, under this restriction is given by:

$$F^*_1(M) = \min_{\{x_1+x_2=\bar{m}_1\}} q^{\bar{m}_1}F^*_2(M - \bar{m}_1) - (V_1 + V_2)q^M + q^{M-\bar{m}_1}[V_1q^{x_1} + V_2q^{x_2}]. \tag{3.29}$$

Note that this can be written as:

$$F^*_1(M) = q^{\bar{m}_1}F^*_2(M - \bar{m}_1) - (V_1 + V_2)q^M + q^{M-\bar{m}_1}\left\{\min_{\{x_1+x_2=\bar{m}_1\}}[V_1q^{x_1} + V_2q^{x_2}]\right\} \tag{3.30}$$

Now notice that the optimal values of the decision variables $x_1$ and $x_2$ can be obtained by solving the corresponding static problem with $\bar{m}_1$ weapons. This complete the proof of the lemma. □

Therefore, if we denote the optimal cost for the static version of the problem with $M$ weapons by $F_s(M)$ then, if the number of weapons to be used in the first stage is restricted to $\bar{m}_1$, we have:

$$F^*_1(M) = q^{\bar{m}_1}F^*_2(M - \bar{m}_1) - (V_1 + V_2)q^M + q^{M-\bar{m}_1}F^*_s(\bar{m}_1). \tag{3.31}$$

Similarly we can restrict the number of weapons to be used in stage 2 to $\bar{m}_2$ and write $F^*_2$ in terms of $F^*_3$. This value can then be substituted in 3.31. This process can be repeated for all stages. If we denote the optimal cost given that the weapons per stage is restricted to $\vec{m}$ by $F_1(M,\vec{m})$ then we obtain (by induction)

$$F_1(M,\vec{m}) = \sum_{t=1}^{T} q^{M-m_t} F_s^*(m_t) - (T-1)(V_1+V_2)q^M. \tag{3.32}$$

The optimal number of weapons per stage can now be obtained by minimizing $F_1(M,\vec{m})$ over $\vec{m}$. Given these optimal values, lemma 3.1 may then be used to obtain the optimal assignments. The optimal solution can therefore be obtained by solving the following problem:

$$F_1^*(M) = \min_{\vec{m}} F_1(M,\vec{m}), \tag{3.33}$$

$$\text{subject to} \quad \sum_{t=1}^{T} m_t = M$$

We will solve this problem by applying theorem B.1. To do this we need to show that the function $F_1(M,\vec{m})$ satisfies the conditions required to apply the theorem. This can be done by showing that the function $q^{M-m_t}F_s^*(m_t)$ is convex. We will state this result as a lemma.

**Lemma 3.2** *The function $q^{M-m_t}F_s^*(m_t)$ is a convex function of $m_t$.*

*Proof:* The proof of this lemma is straightforwarward but lengthy. We have included it in Appendix B. □

If we apply theorem B.1 we will find that the optimal solution has the property that the weapons are spread as evenly as possible among the $T$ stages. Therefore, an optimal strategy for the present stage is to use $\lfloor \frac{M}{T} \rfloor$ of the weapons and to assign these weapons as they would be assigned for the corresponding static problem. This completes the proof. ■

Theorem 3.7 is an interesting result because we find that the weapons are spread evenly among the stages. We will now compute the cost of the optimal strategy. Define the following variables:

$$m_l \stackrel{\text{def}}{=} \lfloor \tfrac{M}{T} \rfloor,$$
$$m \stackrel{\text{def}}{=} \lceil \tfrac{M}{T} \rceil,$$

$$x_{ll} \stackrel{\text{def}}{=} \lfloor \tfrac{m_l}{T} \rfloor,$$
$$x_{lu} \stackrel{\text{def}}{=} \lfloor \tfrac{m_u}{T} \rfloor,$$
$$x_{ul} \stackrel{\text{def}}{=} \lceil \tfrac{m_l}{T} \rceil,$$
$$x_{uu} \stackrel{\text{def}}{=} \lceil \tfrac{m_u}{T} \rceil$$

Using the results from theorem 3.7 it can be shown that the optimal cost $F_1^*(M)$ for the case $V_1 = V_2 = 1$ is given by:

$$F_1^*(M) = (M - Tm_l)[q^{x_{lu}} + q^{x_{uu}}] + (Tm_u - M)[q^{x_{ll}} + q^{x_{ul}}] - 2(T-1)q^M. \tag{3.34}$$

In the special case in which $M = 2kT$ for some positive integer $k$ (i.e. if the two targets survive all stages then in each stage $k$ weapons will be assigned to each of them), the optimal cost can be simplified to

$$F_1^*(M) = 2[Tq^{M-k} - (T-1)q^M]. \tag{3.35}$$

Note that if the number of stages is large then $F_1^*(M) \approx 2q^M$. The optimal cost for the static problem with $M$ weapons is $2q^{\frac{M}{2}}$. This implies that roughly half as many weapons are required for the dynamic case to produce the same optimal cost as for the static one. The comparison of the dynamic and static strategies in terms of the number of weapons saved wil discussed in more detail in subsection 3.3.3.

Let us note the optimal cost for the dynamic two-stage problem using $M$ weapons by $F_d^*(M)$. Let us denote the optimal cost for the corresponding static problem by $F_s^*(M)$. In figure 3.7 we have plotted the log of the ratio of the optimal two-stage cost to the optimal static cost, $\log[F_d^*(M)/F_s^*(M)]$, for different values of total weapons $M = 4, 8, 12, 16, 20$, versus the kill probability $p$. Note that the ratio decreases as $p$ increases as well as when $M$ increases. This means that as either the number and/or the effectiveness of the defensive weapons increase, the advantage of using a two-stage dynamic "shoot-look-shoot-..." type of strategy also increases.

Let us denote the optimal cost for the dynamic $T$-stage problem by $F^*(T)$. In figure 3.8 we have plotted the log of the ratio of the optimal dynamic and static costs $\log[F^*(T)/F^*(1)]$

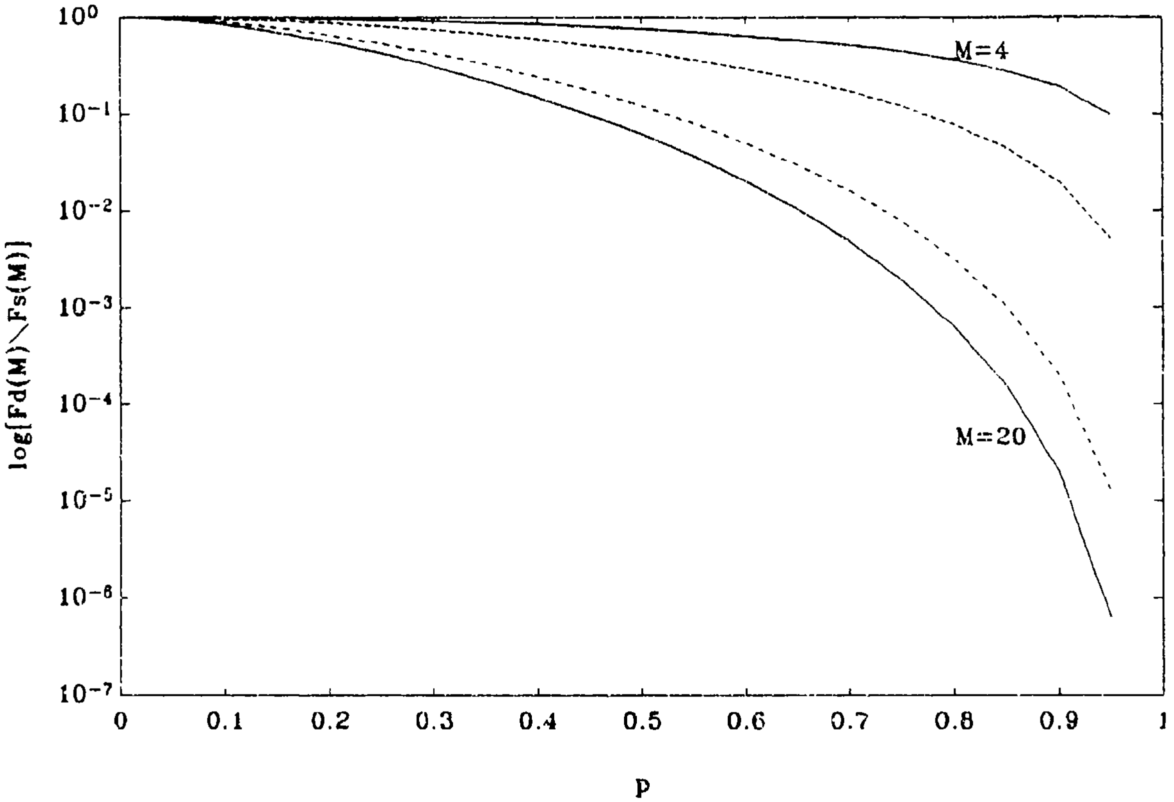


Figure 3.7: Log of the ratio of the optimal two-stage dynamic cost and the optimal static cost vs. the kill probability $p$ for different weapon totals, $M = 4, 8, 12, 16, 20$, $(N = 2)$.

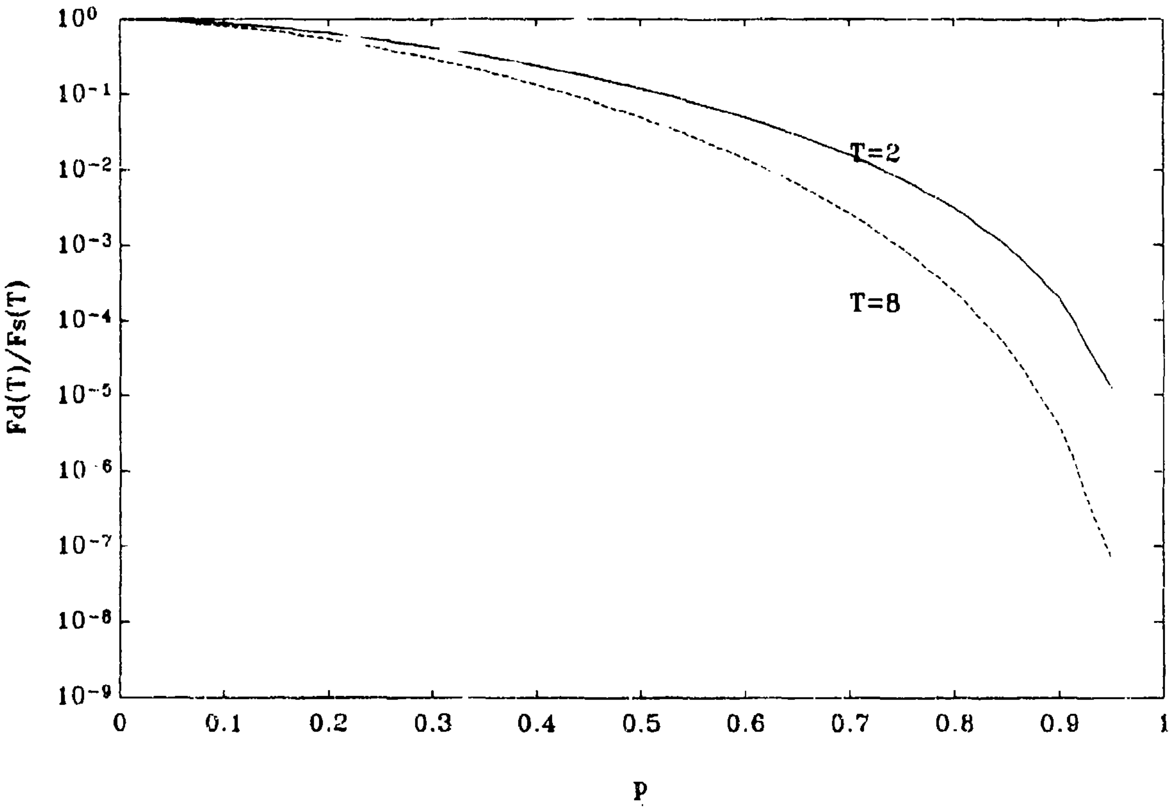


Figure 3.8: Log of the ratio of the optimal $T$-stage dynamic cost and the optimal static cost vs. the kill probability $p$ for different stages $T = 2, 4, 8$ with $M = 16$ and $N = 2$.

versus the kill probability $p$ for different number of stages $T = 2, 4, 8$. The number of weapons is kept fixed at $M = 16$. It can be shown that for $T \geq M/2$ the optimal dynamic cost remains constant. In other words, the inclusion of additional stages does not improve the optimal cost. Note that the cost improvement of the dynamic model improves with the number of stages (up to a point) and with the kill probability.

### 3.3.2 Numerical Results

In this section we will consider the case of $N$ equally valued targets with a uniform kill probability for all weapon-target pairs and all stages. In the previous subsection we considered the case of two targets $N = 2$. In section 3.2 we considered the case in the limit of an infinite number of targets $N \approx \infty$. For general $N$ there does not appear to be an analytic solution to the problem. One must therefore compute solutions numerically. In this section we will compute the solutions for some simple cases and use the results of the previous sections to provide bounds.

Theorem 3.1 states that, in the optimal strategy of this problem, the weapons to be used in each stage should be spread evenly among the surviving targets. The decision variable will therefore be the number of weapons to be used in the first stage, $m_1$. The remaining weapons are used in stage two. Given the optimal values of $m_1$ the optimal assignment can be obtained by spreading these weapons evenly among the targets. The expected cost for the $T$ stage problem in which $m_1$ weapons are used in the first stage will be denoted by $F_1(m_1)$. We computed optimal solutions for a two stage problem with $N$ unit-valued targets, $M$ weapons and a single kill probability $p$ for all weapon-target pairs and both stages.

Table 3.3 contains the optimal values of $m_1$ for the cases $p = 0.9, M = 2, \ldots, 25$, and $N = 2, \ldots, 10$. The cases for which $M$ is a multiple of $N$ is written in boldface type. An interesting feature to note is that if $N \leq M < 2N$ then the optimal value of $m_1$ is $N$. Note that this is not true for $M = 2N$ as can be seen from the case $M = 14, N = 7$. For $M > 2N$ the optimal values of $m_1$ are close to $M/2$ but tend to be a multiple of $N$.

| $M$ | N = 2 | N = 3 | N = 4 | N = 5 | N = 6 | N = 7 | N = 8 | N = 9 | N = 10 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 2 | **3** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | **2** | 3 | **4** | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 2 | 3 | 4 | **5** | 5 | 5 | 5 | 5 | 5 |
| 6 | **3** | **3** | 4 | 5 | **6** | 6 | 6 | 6 | 6 |
| 7 | 4 | 3 | 4 | 5 | 6 | **7** | 7 | 7 | 7 |
| 8 | **4** | 4 | **4** | 5 | 6 | 7 | **8** | 8 | 8 |
| 9 | 5 | **5** | 5 | 5 | 6 | 7 | 8 | **9** | 9 |
| 10 | **5** | 6 | 4 | **5** | 6 | 7 | 8 | 9 | **10** |
| 11 | 6 | 6 | 7 | 5 | 6 | 7 | 8 | 9 | 10 |
| 12 | **6** | **6** | **8** | 6 | **6** | 7 | 8 | 9 | 10 |
| 13 | 7 | 7 | 8 | 7 | 7 | 7 | 8 | 9 | 10 |
| 14 | **7** | 8 | 8 | 10 | 6 | **8** | 8 | 9 | 10 |
| 15 | 8 | **9** | 9 | **9** | 6 | 7 | 8 | 9 | 10 |
| 16 | **8** | 9 | **8** | 10 | 12 | 7 | **8** | 9 | 10 |
| 17 | 9 | 9 | 8 | 10 | 11 | 8 | 8 | 9 | 10 |
| 18 | **9** | **9** | 12 | 10 | **12** | 14 | 8 | **9** | 10 |
| 19 | 10 | 11 | 12 | 10 | 12 | 13 | 10 | 9 | 10 |
| 20 | **10** | 12 | **12** | **10** | 12 | 14 | 8 | 9 | **10** |
| 21 | 11 | **12** | 12 | 11 | 12 | **14** | 15 | 9 | 10 |
| 22 | **11** | 12 | 12 | 10 | 12 | 14 | 16 | 10 | 10 |
| 23 | 12 | 13 | 13 | 15 | 12 | 14 | 16 | 11 | 11 |
| 24 | **12** | **14** | **12** | 14 | **12** | 14 | **16** | 18 | 12 |
| 25 | 13 | 15 | 15 | **15** | 13 | 14 | 16 | 18 | 10 |

Table 3.3: Optimal number of first stage weapons for a two stage problem with a uniform kill probability of $p = 0.9$.

| $M$ | N = 2 | N = 3 | N = 4 | N = 5 | N = 6 | N = 7 | N = 8 | N = 9 | N = 10 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 2 | **3** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | **2** | 3 | **4** | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 2 | 3 | 4 | **5** | 5 | 5 | 5 | 5 | 5 |
| 6 | **3** | **3** | 4 | 5 | **6** | 6 | 6 | 6 | 6 |
| 7 | 4 | 3 | 4 | 5 | 6 | **7** | 7 | 7 | 7 |
| 8 | **4** | 4 | **4** | 5 | 6 | 7 | **8** | 8 | 8 |
| 9 | 5 | **5** | 5 | 5 | 6 | 7 | 8 | **9** | 9 |
| 10 | **5** | 6 | 4 | **5** | 6 | 7 | 8 | 9 | **10** |
| 11 | 6 | 6 | 7 | 5 | 6 | 7 | 8 | 9 | 10 |
| 12 | **6** | **6** | **8** | 6 | **6** | 7 | 8 | 9 | 10 |
| 13 | 7 | 7 | 8 | 7 | 7 | 7 | 8 | 9 | 10 |
| 14 | **7** | 8 | 8 | 10 | 6 | **7** | 8 | 9 | 10 |
| 15 | 8 | **9** | 9 | **10** | 9 | 7 | 8 | 9 | 10 |
| 16 | **8** | 9 | **8** | 10 | 10 | 7 | **8** | 9 | 10 |
| 17 | 9 | 9 | 11 | 10 | 11 | 9 | 8 | 9 | 10 |
| 18 | **9** | **10** | 12 | 10 | **12** | 12 | 8 | **9** | 10 |
| 19 | 10 | 11 | 12 | 10 | 12 | 13 | 9 | 9 | 10 |
| 20 | **10** | 12 | **12** | **10** | 12 | 14 | 12 | 9 | **10** |
| 21 | 11 | **12** | 12 | 11 | 12 | **14** | 15 | 9 | 10 |
| 22 | **11** | 12 | 12 | 14 | 12 | 14 | 16 | 10 | 10 |
| 23 | 12 | 13 | 13 | 15 | 12 | 14 | 15 | 15 | 11 |
| 24 | **12** | **14** | **12** | 15 | **12** | 14 | **16** | 16 | 12 |
| 25 | 13 | 15 | 15 | **15** | 13 | 14 | 16 | 17 | 13 |

Table 3.4: Optimal number of first stage weapons for a two stage problem with a uniform kill probability of $p = 0.5$.

Table 3.4 contains the optimal values of $m_1$ for the cases $p = 0.5, M = 2, \ldots, 25$, and $N = 2, \ldots, 10$. We again find that a good approximation to the optimal value is $m_1^* = N$ for $M \leq 2N$ and $m_1^* \approx \frac{M}{2}$ for $M \geq 2N$. Note that these optimal values do not increase monotonically with the number of weapons as can be seen in the case $N = 4$ for $M = 22, 23, 24$. Here the optimal values of $m_1$ are 12,13,12 respectively. This is due to the discrete nature of the problem.

| $M$ | N = 2 | N = 3 | N = 4 | N = 5 | N = 6 | N = 7 | N = 8 | N = 9 | N =10 |
|---|---|---|---|---|---|---|---|---|---|
| 2 | **2** | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 2 | **3** | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | **2** | 3 | **4** | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 2 | 3 | 4 | **5** | 5 | 5 | 5 | 5 | 5 |
| 6 | **3** | **3** | 4 | 5 | **6** | 6 | 6 | 6 | 6 |
| 7 | 4 | 3 | 4 | 5 | 6 | **7** | 7 | 7 | 7 |
| 8 | **4** | 4 | **4** | 5 | 6 | 7 | **8** | 8 | 8 |
| 9 | 5 | **5** | 4 | 5 | 6 | 7 | 8 | **9** | 9 |
| 10 | **5** | 6 | 4 | **5** | 6 | 7 | 8 | 9 | **10** |
| 11 | 6 | 6 | 5 | 5 | 6 | 7 | 8 | 9 | 10 |
| 12 | **6** | **6** | **6** | 5 | **6** | 7 | 8 | 9 | 10 |
| 13 | 7 | 7 | 7 | 5 | 6 | 7 | 8 | 9 | 10 |
| 14 | **7** | 8 | 8 | 8 | 6 | **7** | 8 | 9 | 10 |
| 15 | 8 | **9** | 8 | **9** | 7 | 7 | 8 | 9 | 10 |
| 16 | **8** | 9 | **8** | 10 | 8 | 7 | **8** | 9 | 10 |
| 17 | 9 | 9 | 8 | 10 | 9 | 7 | 8 | 9 | 10 |
| 18 | **9** | **9** | 9 | 10 | **10** | 8 | 8 | **9** | 10 |
| 19 | 10 | 10 | 11 | 10 | 11 | 9 | 8 | 9 | 10 |
| 20 | **10** | 11 | **12** | **10** | 12 | 10 | 8 | 9 | **10** |
| 21 | 11 | **12** | 12 | 10 | 12 | **11** | 9 | 9 | 10 |
| 22 | **11** | 12 | 12 | 10 | 12 | 12 | 10 | 9 | 10 |
| 23 | 12 | 12 | 12 | 11 | 12 | 13 | 11 | 9 | 10 |
| 24 | **12** | **12** | **12** | 14 | **12** | 14 | **12** | 10 | 10 |
| 25 | 13 | 13 | 13 | **15** | 13 | 14 | 13 | 11 | 10 |

Table 3.5: Optimal number of first stage weapons for a two stage problem with a uniform kill probability of $p = 0.1$.

Table 3.5 are results for the cases $p = 0.1, M = 2, \ldots, 25$ and $N = 2, \ldots, 10$. Note that in most physical situations the kill probability will not be as low as 0.1. However, we wanted to investigate how the optimal values of $m_1$ changed with the kill probability. We found that for $p = 0.9$ and $p = 0.5$ the optimal strategy was to use roughly half of the weapons in stage 1. Therefore if there was a dependency on $p$ we believed that it would show up for the case $p = 0.1$. However, for this case as well we find that the optimal strategy is again to use roughly half of the weapons in stage 1 if $M \geq 2N$.

Observe that, for all cases except one, the optimal value of $m_1$ for the case $M = 2N$ is $N$. Therefore if the weapon-target ratio is 2:1 then it is optimal to use half of the weapons in stage 1. Recall also that for the case $N = 2$ and for large values of $N$ this was the optimal thing to do for a 2:1 weapon-target ratio.

Note that the optimal value of $m_1$ tended to be a multiple of $N$. This tendency seems to be greater for the higher kill probabilities. For example consider the case $M = 25$ and $N = 9$. For a kill probability of 0.9 it is optimal to use 18 weapons in stage 1. Therefore two weapons are fired at each of the targets in stage 1. On the other hand if the kill probability is 0.1 then the optimal strategy is to use 11 weapons in stage 1 which is roughly half of the weapons.

Our conclusion is that it appears that a near-optimal solution to the two-stage problem is to use half of the weapons in stage 1 if the kill probability is small. If the kill probability is large then an approximate solution is to use the number of weapons in stage 1 which is the smallest multiple of $N$ greater than $M/2$.

Figure 3.9 is a plot of the ratio of the optimal dynamic two-stage cost to the optimal static cost versus the kill probability $p$ with a 2:1 weapon to target ratio (i.e $M = 2N$). We have plotted the cases $N$=2,4,6,8 and 10. We have also plotted the ratio in the limit as $N$ goes to infinity. Note that this provides a lower bound for the case of finite $N$. Here we see that, as the sizes of both offensive and defensive stockpiles increase, the cost advantage of the dynamic strategy increases. This implies that, for large-scale problems, the dynamic shoot-look-shoot strategy will have a significant cost advantage over the static one. However, we also note that the increase in the cost advantage decreases with $N$. Therefore, if the number of targets $N$ is very large it might be better to split the problem into two smaller problems. This increases the optimal cost slightly, but greatly reduces the complexity of the problem.

Figure 3.10 contains a plot of the ratio of the optimal two-stage cost to the optimal static cost versus the number of weapons $M$ with a kill probability of $p = 0.5$. We have plotted the cases $N$=2,4,6,8 and 10. Note that the cost advantage of the dynamic strategy increases roughly exponentially with the number of weapons. This implies that the dynamic strategy is significantly better even for relatively small weapon to target ratios.

Figure 3.11 contains a plot of the ratio of the optimal dynamic and static costs versus

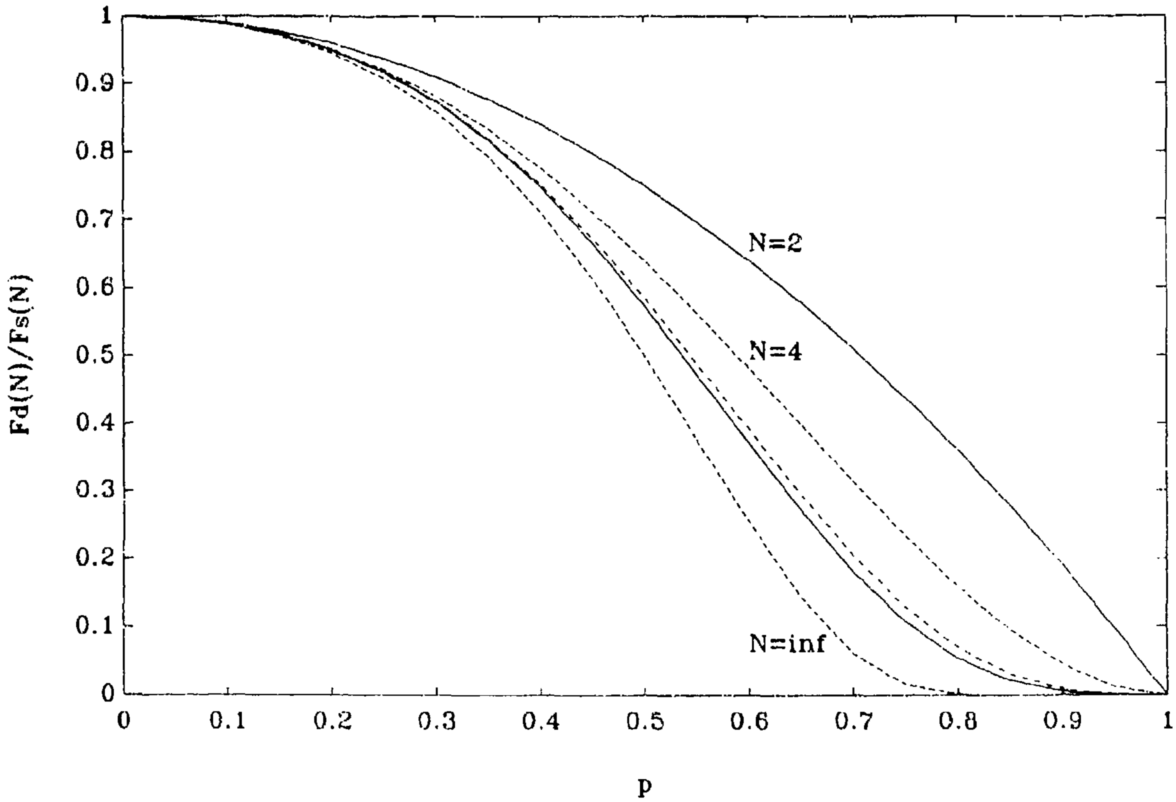


Figure 3.9: Plc' of the ratio of the optimal dynamic (two-stage) and static costs vs the kill probability for a 2:1 weapon-target ratio, ($N = 2, 4, 6, 8, 10, \infty$).

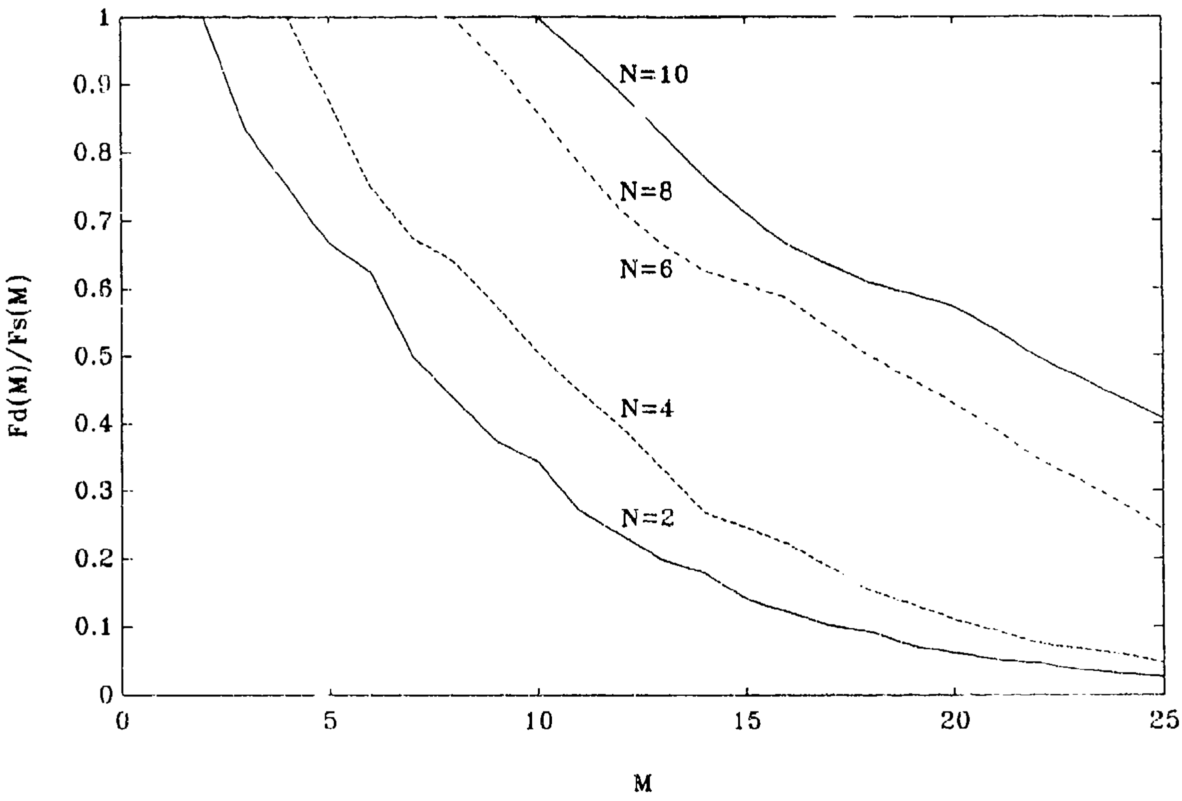


Figure 3.10: Plot of the ratio of the optimal dynamic (two-stage) and static costs vs. the number of weapons $M$, for different numbers of targets $N$=2,4,6,8,10, with $p = 0.5$.

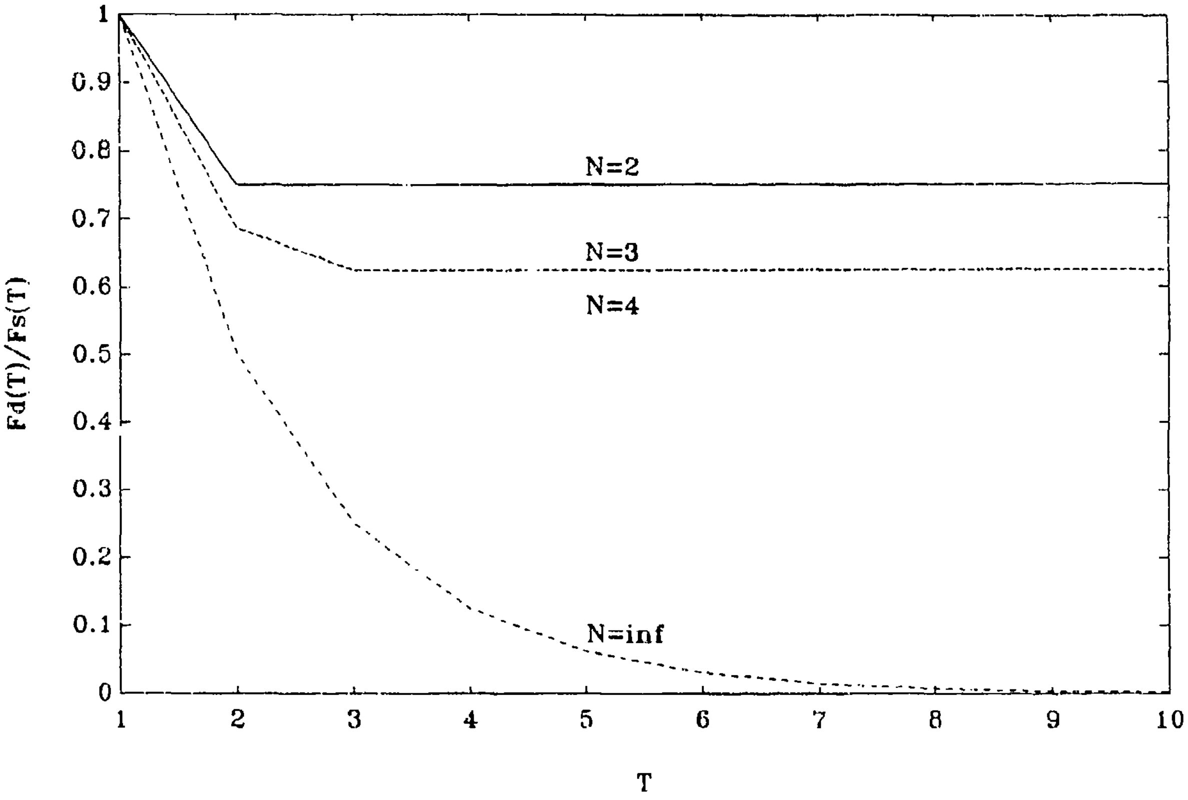


Figure 3.11: Plot of the ratio of the optimal dynamic and static costs vs the number of stages, $T$, available; $M/N = 2$ and $p = 0.5$, $(N = 2, 3, 4, \infty)$.

the number of stages $T$. We used a 2:1 weapon to target ratio and $p = 0.5$. The cases $N = 2, 3$ and 4 were plotted as well as the limiting case as $N$ goes to infinity. The latter plot provides a lower bound for all cases of finite $N$. Using theorem 3.5 we can show that in the limit as $N$ goes to infinity the ratio of the $T$ stage cost to the static cost is equal to $2^{1-T}$.

Note that the advantage of the dynamic strategy increases with the number of stages. For finite values of $N$ the advantage increases up to a finite number of stages. Beyond this point the advantage remains constant because there are not enough weapons to make use of the additional stages. Note that, for large values of $N$, most of the improvement is obtained for a small number of stages (approximately 5 for this example). For a kill probability of 0.8 most of the improvement will be obtained for three stages. Recall that the computational complexity of the problem increases exponentially with the number of stages. This suggests that the defense should use a small number of stages in its strategy (roughly 3) since this provides a significant increase in performance over the static strategy and the computational complexity is not too great.

The above results indicate that the dynamic strategy offers a significant cost improvement over the static strategy but with an increase in the problem complexity. This improvement increases with the size of the problem as well as with the number of stages. We believe that by using sub-optimal algorithms for the dynamic problem one can obtain a significant cost improvement with only a moderate increase in problem complexity.

### 3.3.3 Comparison of Dynamic and Static Strategies

We have seen that the use of a dynamic strategy can significantly decrease the expected number of surviving targets. However, we have also seen that the complexity of such a strategy is much greater than that of the static strategy. One must therefore decide whether the cost improvement of the dynamic strategy is worth the increase in the problem complexity. This decision is of course dependent on the problem being solved and on the computational resources available for solving it. In this section we will provide another view of this trade-off which can help in making the right decision. We will show that by going to a dynamic strategy the defense can effectively double its arsenal of weapons. In other words, half as many weapons (with the same effectiveness) are required for the dynamic strategy to obtained the same cost performance as the static one.

For a given problem one can further reduce the optimal cost by either increasing the number of stages while keeping the number of weapons fixed or increasing the number of weapons while keeping the number of stages fixed. Suppose that $M$ weapons are available and that the optimal static cost is given by $F_s(M)$. Now consider the situation in which the number of stages $T$ is large[2]. Let $M_d$ be the number of weapons which, if used in a dynamic strategy, results in an optimal cost of $F_s(M)$. In other words, let $F_d(M_d) = F_s(M)$. One can then compare $M$ and $M_d$ to decide on whether to use a dynamic strategy or increase the number of weapons.

In the case of $N = 2$ we have shown (section 3.2.1) that if the number of stages is large and $M = 2T$, then the optimal dynamic cost $F_d(M) \approx 2(1-p)^M$. The optimal static cost is given by $F_s(M) = 2(1-p)^{\frac{M}{2}}$. Therefore in this case $M_d = \frac{M}{2}$. In other words, half as many weapons are required for the dynamic strategy to obtain the same optimal cost as the static one.

Let us now consider the case for large values of $N$. Let $M = \kappa N$ for some $\kappa \in \aleph$. The optimal static strategy is to divide the weapons equally among the targets. Therefore, the optimal static cost is given by

$$F_s(M) = N(1-p)^{\kappa}.$$

Let us now consider the dynamic problem. In theorem 3.3 we showed that if the number of stages is large then the optimal strategy at each stage is to assign a single weapon to each of the surviving targets. If $N$ is large then we know, by theorem 3.5, that the problem can be treated as a deterministic one in which the number of targets in a stage is equal to the expected number of targets that survive the previous stage. If a single weapon is assigned

[2]Recall that increasing the number of stages beyond the number of weapons cannot improve the optimal cost.

to each target in stage 1 then the expected number of surviving targets is $N(1-p)$. If a single weapon is assigned to each target in stage 2 then the expected number of targets which survive stage 2 will be $N(1-p)^2$, etc. Therefore, we can use theorem 3.5 to show that, after $T$ stages

$$\lim_{N\to\infty} \frac{F_d(M_d)}{N} = (1-p)^T.$$

Therefore if $N$ is large then $F_d(M_d) \approx N(1-p)^T$. We now need to compute the number of weapons used. Again, for large $N$ it can be shown that

$$\begin{aligned} M_d &\approx N(1+(1-p)+\cdots+(1-p)^{T-1}) \\ &= N\left[\tfrac{1-(1-p)^T}{p}\right] \end{aligned}$$

If we now equate the optimal static cost with the optimal dynamic cost we obtain $T=\kappa$. Therefore

$$\frac{M}{M_d} = \frac{\kappa p}{1-(1-p)^\kappa}$$

We stress that this result is only valid for large values of $N$ and equal valued targets. Let us look at some special cases:

$\kappa = 1$: If $\kappa = 1$ then $\frac{M}{M_d} = 1$. Since $T = \kappa = 1$, this implies that the dynamic strategy is a single stage problem which is the static problem. Therefore, the result is correct.

$p \approx 0$: As $p$ tends to 0, $\frac{M}{M_d}$ converges to 1. Again this is correct since the optimal cost in both cases goes to $N$.

$p \approx 1$: As $p$ tends to 1, $\frac{M}{M_d}$ converges to $\kappa$. Note, however, that if $p = 1$ then the ratio should be 1 since both strategies can destroy all targets with $N$ weapons. If $M = N$ then $\kappa = 1$ so the ratio tends to 1 as required.

$\kappa \approx \infty$: If $\kappa$ is large, so that $(1-p)^{\frac{M}{N}} << 1$, then $\frac{M}{M_d} \approx \kappa p$. Here we see that the defensive advantage grows with the number and effectiveness of the weapons.

Let us consider a typical problem with a 2:1 weapon to target ratio and a kill probability of 0.8. In this case

$$\frac{M}{M_d} = \frac{2}{2-p} = 1.67.$$

Therefore, for typical problems with a large number of targets, roughly half as many weapons are required for the dynamic strategy to produce the same optimal cost as the static one. Recall that the same was found to be true for the case of two targets. We can therefore conclude that for most problems the ratio is roughly half.

## 3.4 Weapon Independent Kill Probabilities

In this section we will study the Two-stage Target-Based problem with the sole restriction that the kill probabilities are independent of the weapons. In chapter 2 we showed that the MMR algorithm produced the optimal assignment for the static version of this problem.In this section we will find that a MMR-type algorithm produces the optimal solution for the case of two targets.

The following notation will be used for the problem. The definitions of all additional notation can be found in Appendix A.

$N \stackrel{\text{def}}{=}$ the number of targets (offense weapons),

$M \stackrel{\text{def}}{=}$ the number of defense weapons,

$V_i \stackrel{\text{def}}{=}$ the value of target $i$, $\quad i = 1, 2, \ldots, N$,

$p_i(1) \stackrel{\text{def}}{=}$ the kill probability of a weapon on target $i$ in stage 1,

$p_i(2) \stackrel{\text{def}}{=}$ the kill probability of a weapon on target $i$ in stage 2,

$m_1 \stackrel{\text{def}}{=}$ the number of weapons used in stage 1,

$m_2 \stackrel{\text{def}}{=}$ the number of weapons used in stage 2, $(m_1 + m_2 = M)$,

$x_i \stackrel{\text{def}}{=}$ the number of weapons assigned to target $i$ in stage 1,

$\vec{x} \stackrel{\text{def}}{=}$ the $N$ dimensional vector $[x_1, \ldots, x_N]^{\mathrm{T}}$.

The vector $\vec{u} \in \{0,1\}^N$ will be used to represent the state of the targets after the first stage.

The states will be denoted as follows:

$$u_i \equiv \begin{cases} 1 & \text{if target } i \text{ survives stage 1} \\ 0 & \text{if target } i \text{ is destroyed in stage 1} \end{cases}$$

If the state of the targets after the first stage is $\vec{u}$, and the number of weapons available for the second stage is $m_2$, the optimal cost for the second stage will be denoted by $F_2(m_2, \vec{u})$.

For the dynamic problem two things must be decided, (a) the optimal number of weapons to be used in the first stage, $m_1$, and (b) the optimal assignment of these weapons to the targets, $\vec{x}$. Note that, given the number of weapons to be used in the first stage and the assignment of these weapons to targets, one can compute the expected cost, under the assumption that the optimal assignment is used in the second stage. Let $F_1(m_1, \vec{x})$ denote the expected cost if $m_1$ weapons are used in the first stage with a first stage assignment given by $\vec{x}$. We have:

$$F_1(m_1, \vec{x}) = \sum_{\{\vec{u} \in \{0,1\}^N\}} \left\{ \prod_{i=1}^{N} \{u_i q_i(1)^{x_i} + (1 - u_i)(1 - q_i(1)^{x_i}\} \right\} F_2(M - m_1, \vec{u}). \qquad (3.36)$$

We can now use this expression to state the Dynamic Target-Based problem in the case of a single class of weapons.

---

**Problem 3.2** *The Single Weapon Class, Dynamic Target-Based (SDTB) problem can be stated as:*

$$\min_{\{m_1 \in Z_+\}} \left\{ \min_{\{\vec{x} \in Z_+^N\}} F_1(m_1, \vec{x}) \right\}$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_i = m_1,$$

$$\text{and} \quad 0 \le m_1 \le M.$$

---

Note that the optimization problem is made up of (a) finding the optimal number of weapons to be used in stage 1 (i.e $m_1^*$) and (b) finding the optimal assignment of the stage one weapons. We will call problem (a) the main problem and problem (b) the assignment subproblem. These problems will be studied separately. If we fix the number of first-stage weapons $m_1$ to be arbitrary, then the assignment subproblem can be written as:

**Problem 3.3** *(Assignment Subproblem):*

$$\min_{\{\vec{x}\in Z_+^N\}} F_1(m_1, \vec{x})$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_i = m_1,$$

Now if we denote the optimal assignment for this subproblem by $\vec{x}^*(m_1)$, then the dynamic problem 3.2 can be restated as follows:

**Problem 3.4** *(Main problem):*

$$\min_{\{m_1\in Z_+\}} F_1(m_1, \vec{x}^*(m_1))$$

$$\text{subject to} \quad 0 \leq m_1 \leq M.$$

In subsection 3.4.1 we will present an MMR-type algorithm for the assignment subproblem. We will show that this algorithm produces an optimal solution for the case of two targets. We have shown that the objective function of problem 3.4 is not unimodal. In subsection 3.4.2 we will assume that the assignment subproblem can be solved and present an algorithm for obtaining at least a local minimum for problem 3.4. The algorithms presented in sections 3.4.1 and 3.4.2 can be combined to obtain an algorithm for producing a near-optimal solution to the original problem 3.2.

### 3.4.1 A MMR Algorithm for the Assignment Subproblem

In this section we will consider the assignment subproblem 3.3. We will assume that $m_1$ is fixed and consider the problem of finding the optimal assignment for these weapons. A

Maximum Marginal Return Algorithm, similar to that used for the static problem, will be presented. We will prove that this algorithm yields optimal assignments for the case of two targets. We conjecture that it produces a near-optimal solution for the case of more than two targets.

The algorithm to be presented, sequentially assigns the $m_1$ weapons in the first stage. Note that in the second stage, $M - m_1$ weapons will be assigned. Let us denote the stage 1 assignment vector if $k$ weapons have been assigned (in stage 1) by $\vec{x}(k)$. The marginal return of adding an additional weapon to target $i$ can then be written as

$$\Delta_i(\vec{x}(k)) = F_1(\vec{x}(k)) - F_1(\vec{x}(k) + e_i)$$

where $e_i^T = (0, \ldots, 0, 1, 0, \ldots, 0)$. For a first stage assignment of $\vec{x}$, let

$S_i(\vec{x}(k)) \stackrel{\text{def}}{=}$ the expected cost, given that target $i$ survives the first stage,

$D_i(\vec{x}(k)) \stackrel{\text{def}}{=}$ the expected cost given that target $i$ is destroyed in the first stage.

Note that $S_i$ and $D_i$ depend on the number of weapons $k$ that have already been assigned in stage 1 as well as the assignment of these weapons $\vec{x}(k)$. Note that it also depends on the number of weapons available in stage 2. It should be emphasized that it was assumed that $m_1$ weapons are available in stage 1 and so $M - m_1$ (*and not* $M - k$) weapons will be assigned in stage 2. We can write

$$F_1(\vec{x}(k)) = q_i(1)^{x_i} S_i(\vec{x}(k)) + (1 - q_i(1)^{x_i}) D_i(\vec{x}(k)),$$

and

$$F_1(\vec{x}(k) + e_i) = q_i(1)^{x_i+1} S_i(\vec{x}(k)) + (1 - q_i(1)^{x_i+1}) D_i(\vec{x}(k)).$$

Therefore

$$\Delta_i(\vec{x}(k)) = p_i(1)(1 - p_i(1))^{x_i} [S_i(\vec{x}(k)) - D_i(\vec{x}(k))].$$

Recall that for the static problem the marginal return of adding a weapon to a target which already had $x_i$ weapons assigned to it was $V_i p_i (1 - p_i)^{x_i}$. Therefore the marginal return for the dynamic problem can be thought of as the marginal return for a static problem with a

```
procedure MMR
Pick a value for m_1;
begin
    x := [0, ..., 0]^T;
    for i = 1:N do Δ_i := Ṽ_i(x(0))p_i(1)
    for j = 1:m_1 do
    begin
        Let r be such that Δ_r = max_i{Δ_i};
        x_r := x_r + 1;
        for i = 1:N do Δ_i := Ṽ_i(x(j))p_i(1)(1 - p_i(1))^{x_i(j)};
    end
end
```

Figure 3.12: A MMR algorithm for the Assignment Subproblem

modified value $\tilde{V}_i(\vec{x}(k))$ given by

$$\tilde{V}_i(\vec{x}(k)) \equiv S_i(\vec{x}(k)) - D_i(\vec{x}(k)).$$

However, note that, unlike the static problem this value depends on the number of weapons $k$ that has already been assigned and it also depends on the specific assignment $\vec{x}$ of these weapons.

In figure 3.12 we have written the code for an MMR algorithm for the assignment subproblem. Each time a weapon is assigned, in stage one, the marginal returns of all targets must be updated. If $k$ weapons have already been assigned in the first stage with an assignment $\vec{x}(k)$, then, the marginal return of increasing the number of weapons assigned to target $i$ is given by

$$\begin{aligned}\Delta_i(\vec{x}(k)) &= p_i(1)(1-p_i(1))^{x_i(k)}[S_i(\vec{x}(k)) - D_i(\vec{x}(k))] \\ &= \tilde{V}_i(\vec{x}(k))p_i(1)(1-p_i(1))^{x_i(k)}.\end{aligned}$$

The only difference between this algorithm and the one that was given for the static problem is that the value of the target is modified each time a weapon is added. Note that the algorithm for the static problem is the special case of the algorithm presented in 3.12 in which $m_1 = M$ and $m_2 = 0$. For this case it is easily seen that $\tilde{V}_i(\vec{x}(k)) = V_i$ for all values of $k$.

One of the most computationaly intensive parts of the algorithm in figure 3.12 is the computation of $\tilde{V}_i(\vec{x}(k))$ especially for large values of $k$. In practice it would be best to approximate this value. One such approximation has been suggested by Castañon et al. [10]. We will briefly describe this approximation.

Let $\vec{u} \in \{0,1\}^{N-1}$ denote the state after stage 1 of all targets except target $i$. Let us denote the second stage cost if the state is $\vec{u}$ and target $i$ is destroyed by $F_2(m_2, \vec{u})$. Also denote the number of weapons assigned to target $i$ in stage 2 if target $i$ survives and that the state of the other targets is $\vec{u}$ by $x_i(\vec{u})$. We can write

$$\begin{aligned}\tilde{V}_i &= S_i - D_i \\ &= \sum_{\vec{w} \in \{0,1\}^{N-1}} \Pr(\vec{u} = \vec{w})[V_i(1 - p_i(1))^{x_i(\vec{w})} + F_2(m_2 - x_i(\vec{w}), \vec{w}) - F_2(m_2, \vec{w})]\end{aligned}$$

We can assume that the function $F_2(m_2, \vec{u})$ is smooth so that if we denote its marginal return at $m_2 - x_i(\vec{u})$ by $\Delta_i(\vec{u})$, then

$$F_2(m_2 - x_i(\vec{u}), \vec{u}) - F_2(m_2, \vec{u}) \approx x_i(\vec{u})\Delta_i(\vec{u}).$$

Therefore

$$\tilde{V}_i \approx \sum_{\vec{w} \in \{0,1\}} \Pr(\vec{u} = \vec{w})[V_i(1 - p_i(1))^{x_i(\vec{w})} + x_i(\vec{w})\Delta(\vec{w})].$$

An iterative procedure is then used to approximate the right hand side.

Another approach that can be used to approximate the value of $\tilde{V}_i$ is to use Monte Carlo simulations[3]. Recall that

$$\tilde{V}_i(\vec{x}(k)) \equiv S_i(\vec{x}(k)) - D_i(\vec{x}(k)).$$

We can find approximate values for $S_i$ and $D_i$ as follows. We will simulate the outcomes of all targets except target $i$ as follows. For each target $j$ we will flip a coin. If $x_j$ weapons have been assigned to the target then the success probability of the coin will be $(1 - p_j(1))^{x_j}$. If the flip is successful then we will assume that the target survived. If the flip is unsuccessful then we will assume that the target was destroyed. Suppose that the target state after all

[3] A similar approach will be used for the more difficult problem in chapter 5.

flips have been performed is $\vec{u}$. $S_i$ is approximated by computing the second stage cost under the assumption that target $i$ survived and the target state of the other targets is $\vec{u}$. Call this cost $\hat{S}_i$. $D_i$ is approximated by computing the second stage cost under the assumption that target $i$ is destroyed and the target state of the other targets is $\vec{u}$. Call this cost $\hat{D}_i$. We then have

$$\tilde{V}_i(\vec{x}(k)) \approx \hat{S}_i(\vec{x}(k)) - \hat{D}_i(\vec{x}(k)).$$

Several of these Monte Carlo runs can be performed and the sample mean taken as an approximation to $\tilde{V}_i$.

We believe that the assignment produced by the MMR algorithm is optimal but have not been able to prove this in general. We can however show that it is optimal for the case of two targets. This result, *which holds for any number of stages* is given in the following theorem:

**Theorem 3.8** *Consider the T-stage Dynamic Target-Based problem with two targets and with weapon independent kill probabilities. If the numbers of weapons to be used at each stage is fixed then the optimal assignment of the weapons for the first stage can be found by using a MMR algorithm.*

*Proof:* To simplify the notation we will denote the kill probabilities of the first stage by $p_i$ instead of $p_i(1)$. We will prove the theorem by induction on the stages. Note that it is true for the case of a single stage since this is the static problem for which the MMR algorithm is optimal. Let us assume that the theorem is true for all stages but the first. We will now prove that it holds for the first stage.

The proof of the optimal assignment of the weapons in the first stage will also be by induction. The number of weapons used in the first stage is $m_1$. If $m_1 = 1$, then this weapon should be assigned to the target for which the reduction in cost is maximum. Therefore, the theorem holds for the case $m_1 = 1$. Let us assume that the theorem is true for $m_1 = K - 1$ weapons and consider the case $m_1 = K$. Denote the optimal assignment for the case of $K - 1$ weapons by $\vec{x}$. We will denote the corresponding cost by $F_1(\vec{x})$. The assignment which is the same as $\vec{x}$ except that an additional weapon is assigned to target $i$ will be denoted by $\vec{x}+e_i$, and the corresponding cost will be denoted by $F_1(\vec{x}+e_i)$. Let

$S_i(\vec{x}) \stackrel{\text{def}}{=}$ the expected cost given that target $i$ survives the first stage.

$D_i(\vec{x}) \stackrel{\text{def}}{=}$ the expected cost given that target $i$ is destroyed in the first stage.

We have

$$\begin{aligned} F_1(\vec{x}+e_i) &= (1-p_i)^{x_i+1}S_i(\vec{x}) + (1-(1-p_i)^{x_i+1})D_i(\vec{x}) \\ &= (1-p_i)^{x_i+1}[S_i(\vec{x}) - D_i(\vec{x})] + D_i(\vec{x}). \end{aligned}$$

Since $F_1(\vec{x}) = (1-p_i)^{x_i}[S_i(\vec{x}) - D_i(\vec{x})] + D_i(\vec{x})$ then we can also write:

$$F_1(\vec{x}+e_i) = (1-p_i)F_1(\vec{x}) + p_iD_i(\vec{x})$$

Let $\vec{r}$ be any assignment of the $K$ weapons. Denote the cost of this assignment by $F_1(\vec{r})$. For any target $j$ such that $r_j > 0$, let $\vec{r}-e_j$ denote the assignment which is the same as $\vec{r}$ except that one less weapon is assigned to target $j$. Let $D_j(\vec{r})$ denote the expected cost given that target $j$ is destroyed in the first stage and let $S_j(\vec{r})$ denote the expected cost given that target $j$ survives the first stage. We then have

$$F_1(\vec{r}) = (1-p_j)F_1(\vec{r}-e_j) + p_jD_j(\vec{r}).$$

Therefore, for any pair of targets $i$ and $j$ with $r_j > 0$ we have

$$F_1(\vec{r}) - F_1(\vec{x}+e_i) = F_1(\vec{r}-e_j) - F_1(\vec{x}) + p_i[F_1(\vec{x}) - D_i(\vec{x})] - p_j[F_1(\vec{r}-e_j) - D_j(\vec{r})].$$

From our induction assumption, we know that $F_1(\vec{r}-e_j) \geq F_1(\vec{x})$. This implies that

$$F_1(\vec{r}) - F_1(\vec{x}+e_i) \geq p_i[F_1(\vec{x}) - D_i(\vec{x})] - p_j[F_1(\vec{r}-e_j) - D_j(\vec{r})].$$

If $p_i[F_1(\vec{x}) - D_i(\vec{x})] \geq p_j[F_1(\vec{r}-e_j) - D_j(\vec{r})]$, then the proof is complete. Unfortunately, this is not always the case so let us assume that $p_i[F_1(\vec{x}) - D_i(\vec{x})] < p_j[F_1(\vec{r}-e_j) - D_j(\vec{r})]$. We will also assume that $i$ is chosen so that $p_i \geq p_j$. Note that this can always be done (since we can always choose $i = j$). These assumptions can be combined to obtain

$$(1-p_j)[F_1(\vec{r}-e_j) - D_j(\vec{r})] \geq (1-p_i)[F_1(\vec{x}) - D_i(\vec{x})]. \tag{3.37}$$

The cost difference of the two assignments can also be written as:

$$F_1(\vec{r}) - F_1(\vec{x}+e_i) = (1-p_j)[F_1(\vec{r}-e_j) - D_j(\vec{r})] - (1-p_i)[F_1(\vec{x}) - D_i(\vec{x})] + D_j(\vec{r}) - D_i(\vec{x}).$$

Using the inequality in 3.37 we have:

$$F_1(\vec{r}) - F_1(\vec{x}+e_i) \geq D_j(\vec{r}) - D_i(\vec{x}). \tag{3.38}$$

Let us assume that target $k$ is the one with the largest marginal return, then

$$F_1(\vec{x}+e_k) \leq F_1(\vec{x}+e_i).$$

Using this in 3.38 we have that

$$F_1(\vec{r}) - F_1(\vec{x}+e_k) \geq D_j(\vec{r}) - D_i(\vec{x}). \tag{3.39}$$

If we can prove that for some $i$ and $j$ such that $r_j > 0$ and $p_i \geq p_j$, $D_j(\vec{r}) \geq D_i(\vec{x})$, then we have $F_1(\vec{r}) \geq F_1(\vec{x}+e_k)$ which implies that the assignment obtained using the MMR algorithm is as good as assignment $\vec{r}$. Since assignment $\vec{r}$ was chosen arbitrarily, then we are done. Since we have not used the fact that $N = 2$ the proof would hold for all $N$. Unfortunately, we have not been able to show in general that, for some $i$ and $j$ such that $r_j > 0$ and $p_i \geq p_j$, $D_j(\vec{r}) \geq D_i(\vec{x})$. However, consider the case of two targets. $N = 2$. Since $\vec{x} \neq \vec{r}$ then there must exist a target $j$ for which $r_j > x_j$. For simplicity let us assume that this inequality holds for target 1. Since there are only two targets then

$$D_1(\vec{r}) = V_2(1-p_2(1))^{r_2}(1-p_2(2))^{m_2}$$

and

$$D_1(\vec{x}) = V_2(1-p_2(1))^{x_2}(1-p_2(2))^{m_2}$$

Since $r_1 > x_1$ then it must be that $x_2 > r_2$ which implies that $D_1(\vec{r}) > D_1(\vec{x})$. Hence the theorem is true for the case of two targets. ■

**procedure**
**begin**
  Pick a value for $m_1 \in \{1, 2, \ldots, M-1\}$;
  **Repeatwhile** $0 < m_1 < M$
    **If** $F_1(m_1+1, \vec{x}^*(m_1+1)) < F_1(m_1, \vec{x}^*(m_1))$ then
      $m_1 \longleftarrow m_1 + 1$;
    **elseif** $F_1(m_1-1, \vec{x}^*(m_1-1)) < F_1(m_1, \vec{x}^*(m_1))$ then
      $m_1 \longleftarrow m_1 - 1$;
    **else**
      **quit**
    **endif**
  **end**
**end**

Figure 3.13: Algorithm for finding a local minimum of the function $F_1(m_1, \vec{x}^*(m_1))$, i.e. the optimal value if the number of stage one weapons is fixed at $m_1$.

### 3.4.2 Optimal Number of First Stage Weapons

In this subsection we will assume that the assignment subproblem 3.3 can be solved. Denote the optimal assignment, if $m_1$ weapons are used in stage 1, by $\vec{x}^*(m_1)$. Since the function $F_1(m_1, \vec{x}^*(m_1))$ has multiple local minima, a global search will have to be done to obtain the global minimum. We believe that for most practical purposes a local minimum will suffice[4]. A local minimum can easily be found by the use of a local search algorithm such as the one given in figure 3.13.

In each iteration an assignment subproblem must be solved. Since this requires a great deal of computation it is best to make approximations. The following approximation has been used by Castañon et al [10].

Let $F_1(m_1, m_2)$ denote the optimal cost for the problem if $m_1$ weapons are used in stage 1 and $m_2$ weapons are used in stage 2. Each iteration requires the computation of the following quantity:

$$F_1(m_1, m_2) - F_1(m_1+1, m_2-1).$$

If this quantity is positive then the solution can be improved by reducing the number of

[4] Recall the example given in figure 3.1 and the corresponding discussion.

weapons in stage 2 by one and increasing the number in stage 1 by one. This quantity can be approximated as follows:

$$F_1(m_1,m_2)-F_1(m_1+1,m_2-1) \approx [F_1(m_1,m_2)-F_1(m_1+1,m_2)+F_1(m_1,m_2)-F_1(m_1,m_2-1)]$$

where $F_1(m_1,m_2) - F_1(m_1+1,m_2)$ is the marginal return of adding a weapon to stage 1. If the MMR algorithm of the previous subsection is used to solve the subproblem, then this difference is given by

$$F_1(m_1,m_2) - F_1(m_1+1,m_2) = \max_i\{\bar{V}_i p_i(1)(1-p_i(1))^{x_i(m_1)}\}.$$

The difference $F_1(m_1,m_2-1) - F_1(m_1,m_2)$ is the marginal loss of removing a weapon from stage 2. This quantity is given by

$$F_1(m_1,m_2-1) - F_1(m_1,m_2) = \sum_{\vec{\omega}\in\{\{0,1\}^N} \Pr(\vec{u}=\vec{\omega}) \min_{\{i|x_i(\vec{u})>0\}}\{V_i p_i(1)(1-p_i(1))^{x_i(\vec{u})-1}\}.$$

An iterative process is used to obtain this difference. These approximations are used to compute an approximate value for the quantity $F_1(m_1,m_2) - F_1(m_1+1,m_2-1)$. Similar approximations can be used to compute an approximate value for the quantity $F_1(m_1,m_2) - F_1(m_1-1,m_2+1)$.

## 3.5 Concluding Remarks

The following conclusions about the dynamic Target-Based problem can be drawn from the results of this chapter.

- An optimal solution cannot be obtained for the general problem 3.1 (in practice) because of the computational complexity of the problem.
- Even under the assumption of weapon independent kill probabilities, the problem is still computationally difficult because multiple minima may exist (proven by example). However, we have also found that, if this is the case then the difference in cost between any two local minima is small compared to the cost of either of them. This suggests that each of these local minima corresponds to a near-optimal solution to the problem.

- If we assume weapon independent kill probabilities and assume that the number of weapons to be used in each stage is fixed, the problem is still difficult. The difficulty is due to the fact that the cost-to-go function is not separable with respect to the assignment variables. We can show that for the case of two targets a MMR algorithm is optimal. We conjecture that such an MMR algorithm will produce a near optimal solution for more than two targets.

- For the case of unit valued targets, a single kill probability and many stages we have found that roughly half as many weapons are required for the dynamic strategy to obtain the same performance as the static one. This result was proven for the case of two targets. We have also shown that it holds approximately for large numbers of targets. Our results also show that most of the efficiency of the dynamic problem is obtained by having 3-5 stages.

- In the case of the two-stage problem with a large number of unit-valued targets, stage dependent kill probabilities in the range $0.6 \leq p(1), p(2) \leq 0.9$, and a 2:1 weapon target ratio, it is optimal to use half of the weapons in stage 1. This suggests that, for the more general problem, if the dependency of the kill probabilities on the stage number is small then a good approximate solution can be obtained by assuming stage independent kill probabilities.

There are several directions in which one may continue this research. One conjecture which we were unable to prove is that the MMR algorithm is optimal for the assignment subproblem for the case of more than two targets. We have not considered solving the most general form of the problem (i.e with kill probabilities that depend on the weapon, target and stage), because of its difficulty. The algorithms in this chapter can be modified and applied to the general problem. Even if heuristics are used, the computation time of these problems is apt to be great for large-scale problems that must be solved in practice.

In practical settings, it is vital that solutions be obtained as quickly as possible. This will require the use of parallel computers. These computers will require algorithms which are easily parallelizable. Research into parallel algorithms for the problem is required. Furthermore, in practice the computational centers and information retrieval centers will be geographically dispersed. This suggests the use of distributed algorithms as well.

# Chapter 4

# The Static Asset-Based Problem

In this chapter we will consider the static version of the Asset-Based WTA Problem. In this problem each offensive target is aimed at a valuable asset of the defense. If it is not engaged it destroys the asset with a given probability called the lethality probability. The defense has a number of weapons with which to engage these targets. As before, the weapon-target engagements are stochastic, and are quantified by kill probabilities. Values are assigned to the assets and the defense must assign weapons to targets with the objective of maximizing the expected total value of the surviving assets. Note that each asset may be attacked by several offensive targets. Also, as before, each target may be engaged by several defensive weapons (salvo attacks).

This model may be more applicable to the later stages of a conflict when the destinations of the targets are more precisely known. Note that in order to save an asset the defense must destroy all of the targets aimed for it. Each of these targets must be attacked with enough weapons so as to make the probability that one or more of them survives sufficiently small. However, if this is done, the defense may not have enough weapons to defend all of the assets. Therefore the defense must decide which of the assets should be defended and assign all of its weapons to the defense of these assets. No weapons should be assigned to the targets aimed for the other not-to-be-defended assets. This is known as a preferential defense strategy (see for example Bracken et al. [17]).

In section 4.1 we will give a mathematical statement of the problem. Since this problem is a more general version of the Static Target-Based problem, then it is NP-Complete. In section 4.2 we will consider the special case of the problem in which the kill probability of each weapon-target pair depends solely on the asset to which the target is aimed and the lethality probability of each target-asset pair is dependent solely on the asset. An algorithm yielding suboptimal solutions will be presented for this problem. In section 4.3 we will consider the problem under the assumption of a single class of weapons. We will also provide an algorithm for this problem which yields suboptimal solutions as well as a method for obtaining an upper bound on the optimal value. In section 4.4 we will approximate the Asset-Based problem by a Target-Based one. The solution methods of chapter 1 can then be used to solve the approximate problem. In section 4.5 we will present some sensitivity analysis results.

The key contribution of this chapter is the algorithm for providing suboptimal solutions for the problem under the assumption of target dependent kill and lethality probabilities. This is the best algorithm, that we are aware of, for this special case of the problem. The algorithm also provides a method for obtaining an upper bound on the optimal value of this special case of the problem.

## 4.1 Problem Definition

We will assume that the engagement of a target by a weapon is independent of all other weapons and targets and that the impact of a target on an asset is independent of all other targets and assets. We will also assume that all weapons are committed "simultaneously", i.e. in a single stage. The following notation will be used. The definitions of all additional notation may be found in Appendix A.

$K \stackrel{\text{def}}{=}$ the number of assets of the defense,

$N \stackrel{\text{def}}{=}$ the number of targets (offense weapons),

$M \stackrel{\text{def}}{=}$ the number of defense weapons,

$G_k \stackrel{\text{def}}{=}$ the set of targets aimed for asset $k$, $\quad k = 1, 2, \ldots, K$,

$n_k \stackrel{\text{def}}{=}$ the number of targets aimed for asset $k$, (i.e. $|G_k|$), $k = 1, 2, \ldots, K$,

$W_k \stackrel{\text{def}}{=}$ the value of asset $k$, $\quad k = 1, 2, \ldots, K$,

$p_{ij} \stackrel{\text{def}}{=}$ the probability that weapon $j$ destroys target $i$ if assigned to it,

$i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, M$,

$\pi_i \stackrel{\text{def}}{=}$ the probability that target $i$ destroys the asset to which it is aimed, $i = 1, 2, \ldots, N$.

The decision variables will be denoted by:

$$x_{ij} = \begin{cases} 1 & \text{if weapon } j \text{ is assigned to target } i \\ 0 & \text{otherwise} \end{cases}$$

The probability that targets $i$ is destroyed is given by $1 - \prod_{j=1}^{M}(1 - p_{ij})^{x_{ij}}$. Therefore the probability that asset $k$ survives all targets aimed for it is given by $\prod_{i \in G_k}[1 - \pi \prod_{j=1}^{M}(1 - p_{ij})^{x_{ij}}]$. Hence we can state the problem as:

---

**Problem 4.1** *The Static Asset-Based (SAB) problem can be stated as:*

$$\max_{\{x_{ij} \in \{0,1\}\}} J = \sum_{k=1}^{K} W_k \prod_{i \in G_k} (1 - \pi_i \prod_{j=1}^{M}(1 - p_{ij})^{x_{ij}}),$$

$$\text{subject to} \quad \sum_{i=1}^{N} x_{ij} = 1, \quad j = 1, 2, \ldots, M.$$

---

The objective function is the sum over all assets of the value of the asset times the probability of survival of the asset. The constraint is due to the fact that each weapon can be assigned to only one target.

The solution to problem 4.1 provides us with an assignment of weapons to targets. However, recall that it may be optimal to use a preferential defense strategy, i.e. defend some of the assets and leave the others undefended. This information can be obtained from the solution of problem SAB. From the solution of SAB we can tell which of the assets should be defended and how each of the defended assets should be defended. We will find that the assets which are defended have large values and/or have few targets aimed for them. The assets which have few targets aimed for them will be assigned a small number of weapons per target. If an asset has many targets aimed for it but has such a large value that it is optimal to defend it, then we will find that many weapons per target will be assigned to defend it. This must be done to ensure that the probability, that one or more of the targets aimed for it get through, is small.

If exactly one target is aimed at each of the assets then problem 4.1 can be simplified by assigning the value of the asset to its target and minimizing the expected total value of the surviving targets. This leads to precisely the Target-Based WTA problem, which means that the Target-Based problem is a special case of the Asset-Based problem. Since the Target-Based problem is NP-Complete, then we conclude that the Asset-Based problem is also NP-Complete.

The Asset-Based problem has proven to be significantly more difficult than the Target-Based problem. The difficulty stems from the fact that, unlike the Target-Based problem which had a convex objective function, the objective function of the SAB problem is neither convex nor concave. Even if we assume that the kill probabilities are independent of the weapons, the problem is still difficult. However, it has not yet been proven whether the problem, under the assumption of weapon independent kill probabilities, is NP-Complete or polynomial time solvable. [1]

[1] Recall that an optimal, polynomial time algorithm exists for the Target-Based problem when this assumption is made. Hence the Target-Based problem, under this assumption, is polynomial time solvable.

## 4.2 Asset Dependent Kill and Lethality Probabilities

In this section we will assume that the kill probability of a weapon-target pair is dependent solely on the asset to which the target is aimed. We will also assume that the lethality probability $\pi_k$ of a target is dependent solely on the asset to which the target is aimed. We will denote the kill probability of a weapon on target $i$ by $p_k$, where $k$ is the asset to which target $i$ is aimed. Let us denote the number of weapons that are assigned to target $i$ by $x_i$.

---

**Problem 4.2** *The Static Asset-Based Problem with Asset dependent probabilities (SABA) can be stated as:*

$$\max_{\{x_i \in Z_+\}} J = \sum_{k=1}^{K} W_k \prod_{i \in G_k} (1 - \pi_k (1 - p_k)^{x_i}),$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_i = M.$$

---

The optimal assignment of problem 4.2 has some important properties which we give in the following theorems.

**Theorem 4.1** *If $\vec{x}$ is an optimal assignment for problem 4.2 then*

$$|x_a - x_b| \leq 1, \qquad \forall\, a, b \in G_k, \quad k = 1, \ldots, K.$$

*Proof:* Pick any asset $k$ and assume that $x_a > x_b + 1$ for some pair of targets $a, b \in G_k$. Let $J(\vec{x})$ denote the value of this assignment. Now consider the assignment which is the same as $\vec{x}$ except that a single weapon is removed from target $a$ and assigned to target $b$. If we use the notation $e_i^T = (0, \ldots, 0, 1, 0, \ldots, 0)$, then this assignment can be written as $\vec{x} - e_a + e_b$. We will denote the value of this assignment by $J(\vec{x} - e_a + e_b)$. We have:

$$J(\vec{x}) - J(\vec{x} - e_a + e_b) = W_k \pi_k p_k [(1 - p_k)^{x_a - 1} - (1 - p_k)^{x_b}] \prod_{\substack{i \in G_k \\ i \neq a,b}} (1 - (1 - p_k)^{x_i}) \qquad (4.1)$$

Since $x_a > x_b + 1$ then the right hand side of 4.1 is negative. Therefore,

$$J(\vec{x}) - J(\vec{x} - e_a + e_b) < 0.$$

This is a contradiction since the assignment $\vec{x}$ was assumed to be optimal. ■

This theorem states that, in the optimal assignment, the numbers of weapons assigned to any two targets aimed for the same asset are either equal or differ by one. This result can also be seen by using a symmetry argument. Theorem 4.1 can be used to simplify problem 4.2 by introducing a new decision variable $X_k$ which will be used to denote the number of weapons assigned to all targets against asset $k$. Given $X_k$ one can obtain an optimal assignment by spreading the weapons evenly among the targets. We will let $\vec{X}$ be the $K$-dimensional vector with elements $X_k$. Let us define $J_k(X_k)$ to be the expected surviving value of asset $k$ if $X_k$ weapons are assigned to its targets and these weapons are spread as evenly as possible among the targets. Then,

$$J_k(X) = W_k(1 - \pi_k(1 - p_k)^{\lceil \frac{X}{n_k} \rceil})^{X - n_k \lfloor \frac{X}{n_k} \rfloor}(1 - \pi_k(1 - p_k)^{\lfloor \frac{X}{n_k} \rfloor})^{n_k(1 + \lfloor \frac{X}{n_k} \rfloor) - X} \qquad (4.2)$$

We can now use equation 4.2 to simplify problem 4.2 to

---

**Problem 4.3**

$$\max_{\vec{X} \in Z_+^K} J = \sum_{k=1}^{K} W_k J_k(X_k),$$

$$\text{subject to} \qquad \sum_{k=1}^{K} X_k = M.$$

---

Note that the objective function in problem 4.3 is separable. If each of the functions $J_k$ was concave then theorem B.1 could be applied to show that an MMR algorithm will produce an optimal solution. Unfortunately, the functions $J_k$ are not concave; but they do have some important properties which we will exploit to deduce an algorithm for problem 4.3.

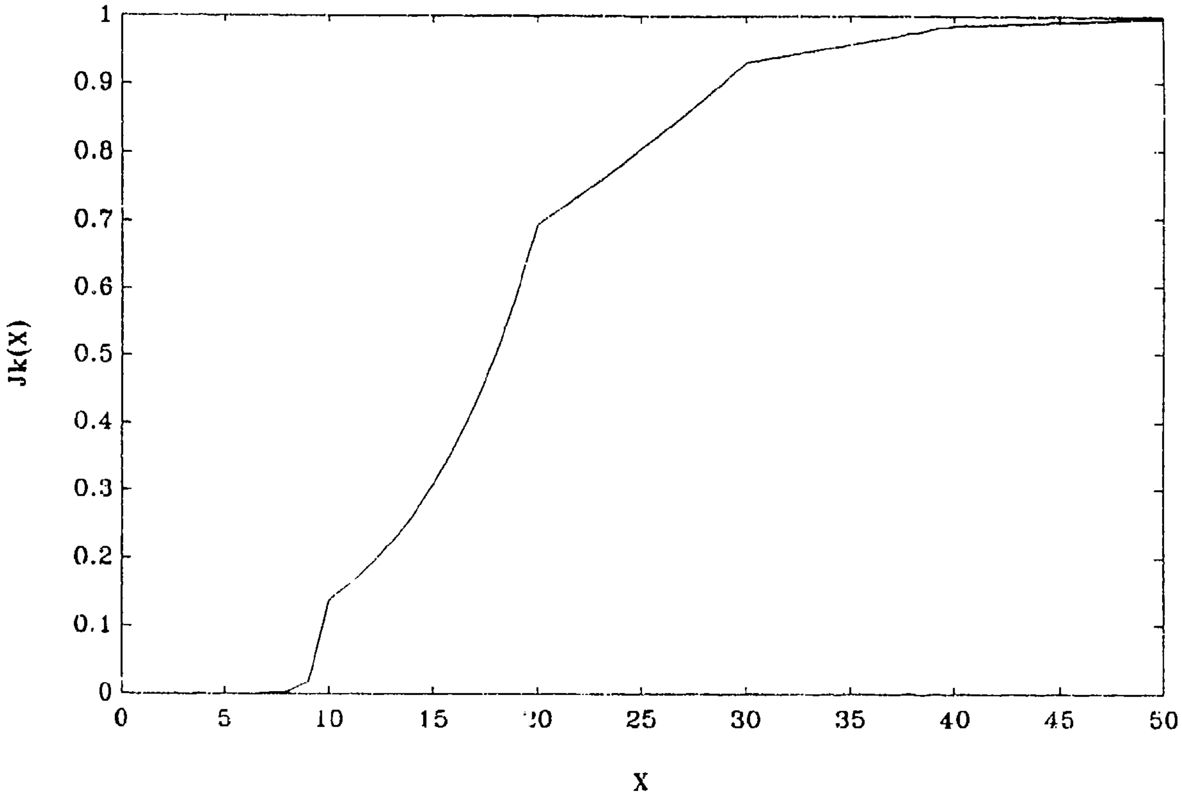


Figure 4.1: An example of the function $J_k$, the expected surviving value of asset $k$, for $n_k = 10, p_k = 0.8, \pi_k = 0.9$ and $W_k = 1$, vs. the total number of weapons assigned to defend it, $X$.

In figure 4.1 we have plotted an example of the function $J_k$ for the case $n_k = 10, p_k = 0.8, \pi_k = 0.9$ and $W_k = 1$. Clearly it is neither convex nor concave. Note that between multiples of $n_k$ the function is convex. This is due to the fact that the weakest link in the defense of the asset is the target to which the least number of weapons is assigned. As a function of multiples of $n_k$, the function is first convex and then becomes concave. This change occurs roughly at the point where the expected number of surviving targets is one i.e. the value of $X$ for which

$$n_k \pi_k (1 - p_k)^{\frac{X}{n_k}} = 1.$$

These two properties can be stated formally as follows.

**Property 1:** If $\lfloor \frac{X}{n_k} \rfloor < X < \lceil \frac{X}{n_k} \rceil$ then

$$J_k(X-1) - 2J_k(X) + J_k(X+1) \geq 0.$$

*Proof:* Let us define

$$\alpha \equiv 1 - \pi_k (1 - p_k)^{\lceil \frac{X}{n_k} \rceil},$$

and

$$\beta \equiv 1 - \pi_k (1 - p_k)^{\lfloor \frac{X}{n_k} \rfloor},$$

Note that

$$J_k(X+1) = J_k(X) \frac{\alpha}{\beta},$$

and

$$J_k(X-1) = J_k(X) \frac{\beta}{\alpha}.$$

Therefore,

$$\begin{aligned} J_k(X-1) - 2J_k(X) + J_k(X+1) &= J_k(X)[\frac{\alpha}{\beta} + \frac{\beta}{\alpha} - 2] \\ &= J_k(X)[\alpha - \beta]^2/(\alpha\beta) \\ &\geq 0, \end{aligned}$$

which completes the proof. ∎

**Property 2:** There exists an $r_k \in Z_+$ such that for all $r \geq r_k$,

$$J_k(n_k r) - 2J_k(n_k(r+1)) + J_k(n_k(r+2)) \leq 0.$$

Furthermore, if $r_k > 0$ then for all $0 < r \leq r_k$,

$$J_k(n_k(r-1)) - 2J_k(n_k r) + J_k(n_k(r+1)) \geq 0.$$

*Proof:* Note that

$$J_k(n_k r) = W_k(1 - \pi_k(1-p_k)^r)^{n_k}.$$

Let $J_k''(n_k r)$ denote the second derivative of $J_k(n_k r)$ with respect to $r$. We have

$$J_k''(n_k r) = W_k n_k \pi_k (1-p_k)^r \log^2(1-p_k)(1-\pi_k(1-p_k)^r)^{n_k-2}[n_k\pi_k(1-p_k)^r - 1].$$

Therefore if $r \leq \left\lfloor \frac{-\log(n_k\pi_k)}{\log(1-p_k)} \right\rfloor$, then $J_k''(n_k r) \geq 0$ and the function is convex. Otherwise if $r \geq \left\lceil \frac{-\log(n_k\pi_k)}{\log(1-p_k)} \right\rceil$ then $J_k''(n_k r) \leq 0$ and the function is concave. ■

The first property says that the function $J_k$ is convex between multiples of $n_k$. The second property says that the function $J_k(rn_k)$ is at first convex and then becomes concave.

If we approximate the function $J_k$ by a concave function then the MMR algorithm, applied to the approximate problem, will produce the optimal solution for the approximate problem. We will approximate each of the functions $J_k$ by its concave hull which we will denote by $\bar{J}_k$. Note that, because of property 1, the line through the origin which is tangent to the function $J_k$ will touch at a point where $X$ is a multiple of $n_k$. Define $\ell_k \in Z_+$ to be such that $X = \ell_k n_k$ is the point at which the tangent through the origin is tangent to $J_k$. Next note that, because of property 2, $\ell_k \geq r_k$ where $r_k$ is the point at which the function $J_k(rn_k)$ changes from being convex to concave. This implies that the function $J_k(rn_k)$ is concave for $r \geq \ell_k$. These facts can now be used to obtain the concave hull $\bar{J}_k$ as

$$\bar{J}_k(X) = \begin{cases} \frac{X}{n_k\ell_k} J_k(n_k\ell_k) & \text{if } X \leq n_k\ell_k \\ J_k(n_k\lfloor \frac{X}{n_k} \rfloor) + (\frac{X}{n_k} - \lfloor \frac{X}{n_k} \rfloor)(J_k(n_k\lceil \frac{X}{n_k} \rceil) - J_k(n_k\lfloor \frac{X}{n_k} \rfloor)) & \text{if } X > n_k\ell_k \end{cases} \tag{4.3}$$

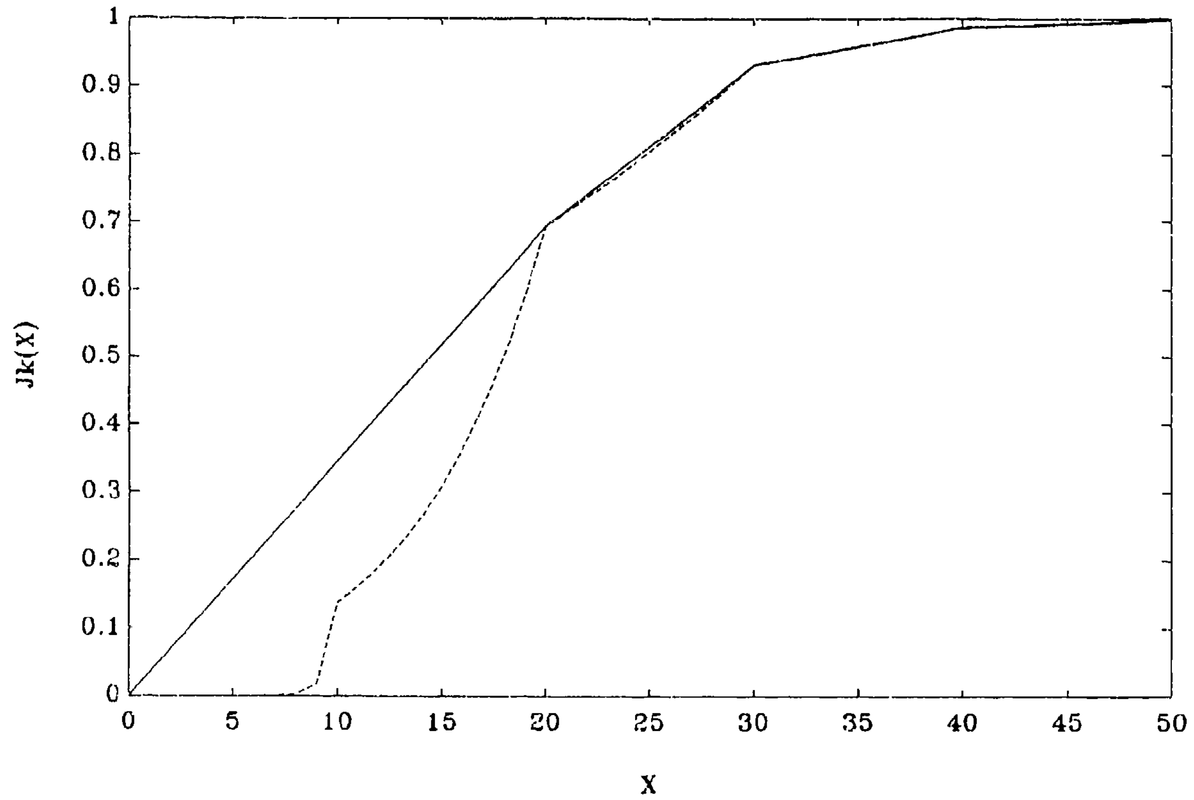


Figure 4.2: The hull $\bar{J}_k(X)$ of the function $J_k(X)$ shown in figure 4.1. The dashed line is $J_k(X)$ and the solid line is $\bar{J}_k(X)$.

In figure 4.2 we have drawn the hull for the function that was plotted in figure 4.1. The dotted line is the function $J_k$ while the solid line is the function $\bar{J}_k$. Note that in this example $\ell_k = 2$ and $r_k = 1$. Secondly note that $\bar{J}_k(X)$ is a good approximation of $J_k(X)$ for $X \geq n_k \ell_k$. This fact will be used in obtaining bounds on the optimal value for the problem.

Consider, now the problem in which we replace the objective function in 4.3 by its concave hull.

---

**Problem 4.4** *An approximation to problem 4.3 is given by:*

$$\max_{\vec{X} \in Z_+^K} \bar{J} = \sum_{k=1}^{K} W_k \bar{J}_k(X_k),$$

$$\text{subject to} \qquad \sum_{k=1}^{K} X_k = M.$$

---

Note that theorem[2] B.1 can be applied to this problem to conclude that the MMR algorithm produces the optimal solution. Denote the optimal solution to 4.4, if an MMR algorithm is used, by $\vec{\bar{X}}^*$. The assignment $\vec{\bar{X}}^*$ has the following important property.

**Property 3:** For all but one of the assets $k$, $\bar{X}_k^*$ is a multiple of $n_k$.

*Proof:* The proof will be by contradiction. Assume that the property does not hold. This means that there exist at least two assets with the property that the total number of weapons assigned to the targets aimed for them is not a multiple of the number of targets aimed for them. For simplicity let us assume that two of these assets are assets 1 and 2. Note that the function $\bar{J}_k$ is linear between multiples of $n_k$. Therefore, the marginal return of asset $k$ is constant between multiples of $n_k$. If the marginal return for asset 1 on termination of the algorithm is greater than that of asset 2 then the weapons that were assigned to asset 2 would have been assigned to asset

[2]Note that in theorem B.1 a convex function is being minimized. The theorem can be applied to problem 4.4 by maximizing the negative of the objective function

1 leading to a contradiction. Therefore the marginal return of asset 2 on termination of the algorithm must be greater than or equal to that of asset 1. If this is the case, then, since the algorithm started assigning weapons to asset 2 then it would have continued doing so until the number of weapons assigned was a multiple of $n_2$ (i.e. until the marginal return for asset 2 changed value). This is a contradiction since we assumed that the number of weapons assigned to asset 2 was not a multiple of $n_2$. ∎

This property states the following. If an asset is defended, then the same number of weapons is assigned to each of the targets aimed for the asset. Because the total number of weapons is arbitrary, then it may not be possible to do this for all of the defended assets. Therefore, the property may not hold for one of the defended assets. Let target $v$ be the target for which $\bar{X}_v^*$ is *not* a multiple of $n_v$, i.e the property is *violated.*

By examining 4.3 one can see that if $X$ is a multiple of $n_k$ then $\bar{J}_k(X) = J_k(X)$. Since $\bar{X}_k^*$ is a multiple of $n_k$ for all assets $k$ except asset $v$ then:

$$\bar{J}(\vec{\bar{X}}^*) = J(\vec{\bar{X}}^*) - J(\bar{X}_v^*) + \bar{J}(\bar{X}_v^*). \tag{4.4}$$

Finally note that $\bar{J}(X)$ is an upper bound to $J(X)$. Therefore, if we denote the optimal solution to the original problem 4.2 by $\vec{X}^*$ then

$$\bar{J}(\vec{\bar{X}}^*) \geq J(\vec{X}^*). \tag{4.5}$$

Furthermore since $\vec{X}^*$ is optimal for Problem 4.2 then

$$J(\vec{X}^*) \geq J(\vec{\bar{X}}^*). \tag{4.6}$$

Combining equations 4.4, 4.5 and 4.6 we obtain

$$J(\vec{\bar{X}}^*) + \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*) \geq J(\vec{X}^*) \geq J(\vec{\bar{X}}^*) \tag{4.7}$$

Therefore the optimal value of problem 4.4, the approximate problem, can be used to obtain upper and lower bounds on the optimal value of problem 4.3.

Notice that the solution to the approximate problem 4.4 is a suboptimal solution to the original problem 4.3. The difference in value between the optimal and suboptimal solutions is bounded by

$$J(\vec{X}^*) - J(\bar{\vec{X}}^*) \leq \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*) \tag{4.8}$$

Note that if $\bar{J}(\bar{X}_v^*) = J(\bar{X}_v^*)$, which would be the case if $\bar{X}_v^*$ is also a multiple of $n_v$ then we obtain

$$J(\vec{X}^*) = J(\bar{\vec{X}}^*)$$

which implies that $\bar{\vec{X}}^*$ is an optimal solution to problem 4.3. In other words, if the total number of weapons is such that for *each* defended asset the same number of weapons is assigned to each of the targets aimed for the asset, then the algorithm produces the optimal solution.

Let us now consider the case in which $\bar{J}(\bar{X}_v^*) > J(\bar{X}_v^*)$. In this case $\bar{X}_v^*$ is not a multiple of $n_v$. By the nature of the MMR algorithm, if the number of weapons is increased by $n_v\lceil\frac{\bar{X}_v^*}{n_v}\rceil - \bar{X}_v^*$ weapons, then the optimal solution to problem 4.4 will be the same except that $\bar{X}_v^*$ will be increased by the number of additional weapons making it a multiple of $n_v$. The analysis in the previous paragraph can then be used to show that the optimal solution for the approximate problem is also optimal for the original problem 4.3. Similarly if the number of weapons is *decreased* by $\bar{X}_v^* - n_v\lfloor\frac{\bar{X}_v^*}{n_v}\rfloor$ then the resulting optimal solution of the approximate problem is optimal for the original problem 4.3 with the decreased number of weapons. These results suggest that the optimal solution obtained for the approximate probiem is close to being optimal for the true problem 4.3. We will now state our result as a theorem.

**Theorem 4.2** *Consider the Static Asset-Based problem in which the kill probability of a weapon-target pair and the lethality probability of a target-asset pair is dependent solely on the asset to which the target is aimed. Let $\bar{\vec{X}}^*$ be the optimal solution to the approximate problem defined in 4.4 obtained by the use of the MMR algorithm. Let $\vec{X}^*$ denote the optimal* *solution of the true problem (i.e problem 4.3) then*

$$J(\bar{\vec{X}}^*) + \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*) \geq J(\vec{X}^*) \geq J(\bar{\vec{X}}^*).$$

*Furthermore if we let*

$$\varepsilon = \max_k \max_{\{0 \leq X_k \leq \ell_k n_k\}} \bar{J}_k(X_k) - J_k(X_k)$$

*then*

$$J(\vec{X}^*) - J(\bar{\vec{X}}^*) \leq \varepsilon.$$

*Proof:* The first part of the theorem has already been proved. The second part is obtained by upper bounding the difference $\bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*)$ by its maximum possible value. ∎

Note that $\varepsilon$ is the maximum over all assets of the maximum difference between the function $J_k$ and its concave hull $\bar{J}_k$. Therefore $\varepsilon$ is dependent solely on the shape of the functions $J_k$. If we increase the size of the problem by in reasing the number of assets, targets and weapons but do not change the types of assets so that $\varepsilon$ does not change then we find that as the problem size increases, the percentage error of the suboptimal solution decreases because $\varepsilon$ remains constant. Therefore, for large-scale problems we expect that the algorithm will perform well.

Note that the bound $\varepsilon$ can be computed even before the problem 4.3 is solved by a MMR algorithm. This provides an upper bound on the error of the suboptimal solution that is obtained by the algorithm. However, after the approximate problem 4.4 has been solved a much tighter bound is obtained by the difference $\bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*)$. Furthermore, as we have shown, the solution can be mad optimal by slightly decreasing or increasing the number of weapons.

## 4.3 A Single Class of Weapons

In the previous section we assumed that the kill and lethality probabilities were solely asset dependent. In this section we will assume that there is a single class of weapons. In other words the kill probabilities are weapon independent and the lethality probability is solely target dependent. These assumptions are valid for the case of a single cluster of weapons. Note that this is a more general problem than that of the previous section. We will present an algorithm which yields a suboptimal solution for this problem which is similar to that presented for problem 4.2. The following notation will be used. The definitions of all additional notation may be found in Appendix A.

$K \stackrel{\text{def}}{=}$ the number of assets of the defense,

$N \stackrel{\text{def}}{=}$ the number of targets (offense weapons),

$M \stackrel{\text{def}}{=}$ the number of defense weapons,

$W_k \stackrel{\text{def}}{=}$ the value of asset $k$ to the defense, $k = 1, 2, \ldots, K$,

$G_k \stackrel{\text{def}}{=}$ the set of targets aimed for asset $k$, $\quad k = 1, 2, \ldots, K$,

$n_k \stackrel{\text{def}}{=}$ the number of targets aimed for asset $k$, $(|G_k|)$, $\quad k = 1, 2, \ldots, K$,

$\pi_i \stackrel{\text{def}}{=}$ the probability that target $i$ destroys the asset to which it is aimed, $i = 1, 2, \ldots, N$,

$p_i \stackrel{\text{def}}{=}$ the probability that a weapon destroys target $i$ if it is fired at it, $i = 1, 2, \ldots, N$.

$x_i \stackrel{\text{def}}{=}$ the number of weapons assigned to target $i$, $\quad i = 1, 2, \ldots, N$,

$\vec{x} \stackrel{\text{def}}{=}$ the $N$-dimensional vector $[x_1, \ldots, x_N]^T$,

$X_k \stackrel{\text{def}}{=}$ the number of weapons assigned to the defense of asset $k$, (i.e. $\sum_{i \in G_k} x_i$),

$\vec{X} \stackrel{\text{def}}{=}$ the $K$ dimensional vector $[X_1, \ldots, X_K]^T$.

Under the assumptions we have made, the probability that an asset survives is the product over all targets of the probability that the target is destroyed. We therefore have:

---

**Problem 4.5** *The Single Weapon Class Static Asset-Based (SSAB) problem can be stated as:*

$$\max_{\vec{x}\in Z_+^N} J(\vec{x}) = \sum_{k=1}^{K} W_k \prod_{i\in G_k} (1 - \pi_i(1-p_i)^{x_i}),$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_i = M.$$

---

The objective function is the total expected surviving asset value and the constraint is due to the fact that the total number of weapons fired must equal the number of weapons available.

Because problem 4.5 is separable with respect to the assets, it can be re-formulated as follows. Let $J_k(X)$ denote the maximum expected surviving value of asset $k$ given that $X$ weapons are used to defend it.

**Problem 4.6** *The subproblem (SUB) is defined by:*

$$J_k(X_k) = \max_{\vec{x}\in Z_+^{n_k}} W_k \prod_{i\in G_k} (1 - \pi_i(1-p_i)^{x_i}),$$

$$\text{subject to} \qquad \sum_{i\in G_k} x_i = X_k.$$

We can now restate the original problem.

**Problem 4.7** *The problem SSAB can be restated as (MAIN):*

$$\max_{\vec{X}\in Z_+^K} J(\vec{X}) = \sum_{k=1}^{K} J_k(X_k)$$

$$\text{subject to} \qquad \sum_{k=1}^{K} X_k = M.$$

We will first consider the subproblems 4.6. The approach will then be the same as in the previous section. We will find the hull of $J_k$ and then use an MMR algorithm on the approximate problem in which $J_k$ is replaced by its hull in problem 4.7. We will then show that the solution of this approximate problem is a near-optimal solution to the true problem 4.7. Since the approach is identical to that used in the previous section, some of the details will be omitted.

### 4.3.1 Solution of the Subproblem

Since the logarithm function is monotonic, if we replace the objective function of problem 4.6 by its logarithm then the optimal assignment of the resulting problem will also be optimal for the original problem. If we take the logarithm of the objective function of SUB we obtain

$$\ln W_k + \sum_{i \in G_k} \ln[1 - \pi_i(1-p_i)^{x_i}].$$

The first term is constant so we can remove it and optimize the second term. The optimization problem is:

$$\max_{\vec{x} \in Z_+^{n_k}} \mathcal{F}(\vec{x}_k) = \sum_{i \in G_k} \ln[1 - \pi_i(1-p_i)^{x_i}], \tag{4.9}$$

$$\text{subject to} \qquad \sum_{i \in G_k} x_i = X_k.$$

Note that the function $\mathcal{F}(\vec{x}_k)$ is separable with respect to the target index $i$. Next note that each of the functions $\ln[1 - \pi_i(1-p_i)^{x_i}]$ is concave. This can be verified by showing that the second derivative of this function (with respect to $x_i$) is non-positive. Therefore, the objective function $\mathcal{F}$ satisifies the conditions required to apply theorem B.1. Hence, a MMR algorithm will produce the optimal solution. This solution will also be optimal for problem 4.6.

We next need to obtain the concave hull of the function $J_k$. In the previous section this task was easy because in that case the functions $J_k$ had two special properties which could be exploited, (a) $J_k$ is convex between multiples of $n_k$ and (b) as a function of multiples of $n_k$ the function is first convex and then concave. In this case however, the functions

$J_k$ do not have these special properties. The functions $J_k$ in this section differ from those of the previous section because the kill probabilities and lethality probabilities are target dependent instead of asset dependent. Let us investigate how these two differences affect the two properties (a) and (b) of the functions $J_k$ in the previous section.

We will first investigate how the convexity of the function $J_k(X)$ between multiples of $n_k$ is affected when the values of the kill and lethality probabilities are allowed to be target dependent. Our intuition implies that the variation in these probabilities should have a smoothing effect on the function $J_k(X)$ which means that the function should be "less convex" between multiples of $n_k$.

We will first consider the effect on the function $J_k(X)$ when the kill probability of a weapon target pair is dependent only on the target. For this case we will assume that the probability that a target destroys an asset is 0.9. We will show the effect for a range of cases which are of importance to us.

The convex region with the largest curvature is the region $n_k \le X \le 2n_k$. We will therefore examine what happens in this region. In order to solve the problem with target dependent kill probabilities we will assume that the range of the kill probabilities is such that the optimal assignment for the case of $2n_k$ weapons is to assign 2 weapons to each of the targets. Some simple calculations show that this is true for example if $0.65 \le p_i \le 0.90$ for all targets $i$. Let the kill probability for the target independent case be $p$ and let the set of kill probabilities for the target dependent case be $\{p_i\}$. We will choose $\{p_i\}$ so that the expected surviving asset value for both cases are the same at $X = n_k$ and at $X = 2n_k$. Let $J_k(X)$ denote the maximum surviving asset value for the case of the target independent kill probabilities and let $\hat{J}_k(X)$ denote it for the case of the target dependent kill probabilities. One can show that $\hat{J}_k(X) \ge J_k(X)$ for $n_k \le X \le 2n_k$. Since $\hat{J}_k(n_k) = J_k(n_k)$ and $\hat{J}_k(2n_k) = J_k(2n_k)$ (by the choice of $\{p_i\}$) then this suggests that in the region $n_k \le X \le 2n_k$ the curvature of the function $\hat{J}_k$ is less than that of the function $J_k$.

Consider the problem of a single asset with $W = 1, n = 10$ and $\pi_i = 0.9$. In figure 4.3 we

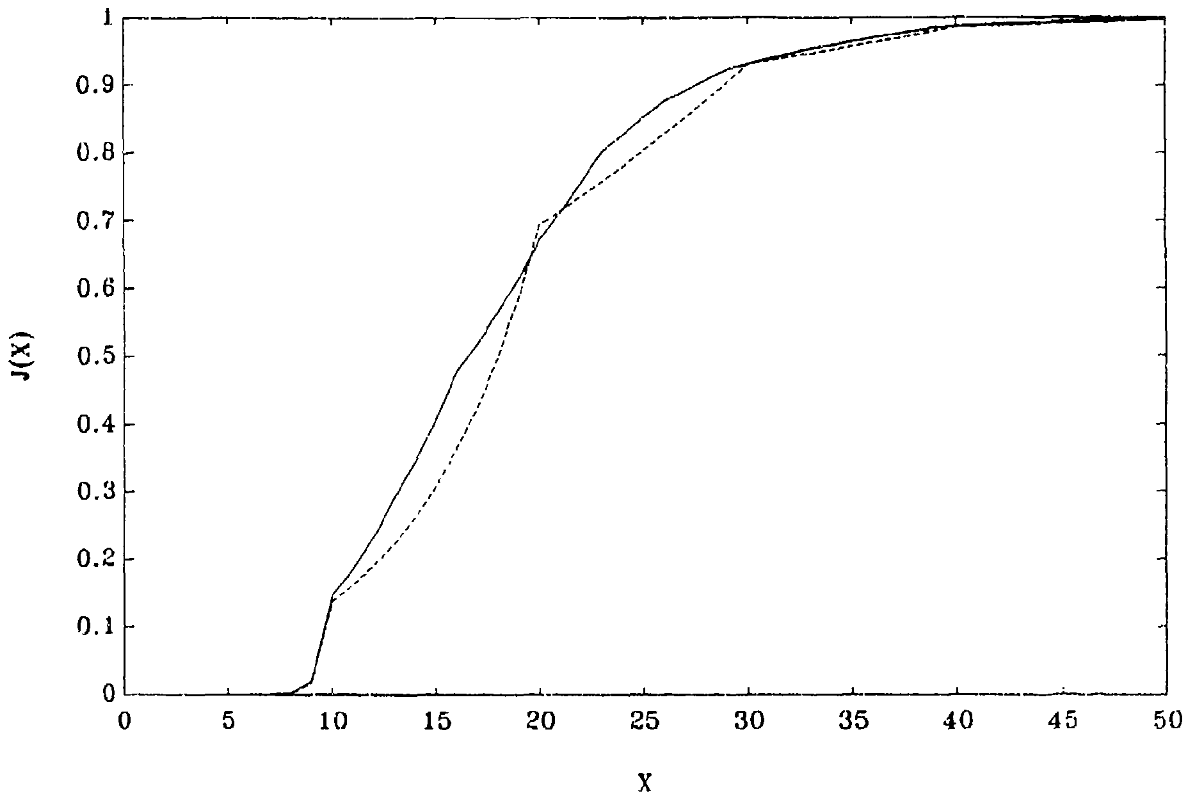


Figure 4.3: Maximum expected surviving value of an asset with $W = 1, n = 10, \pi = 0.9$ for target dependent kill probabilities (solid line) and target independent kill probabilities (dashed line).

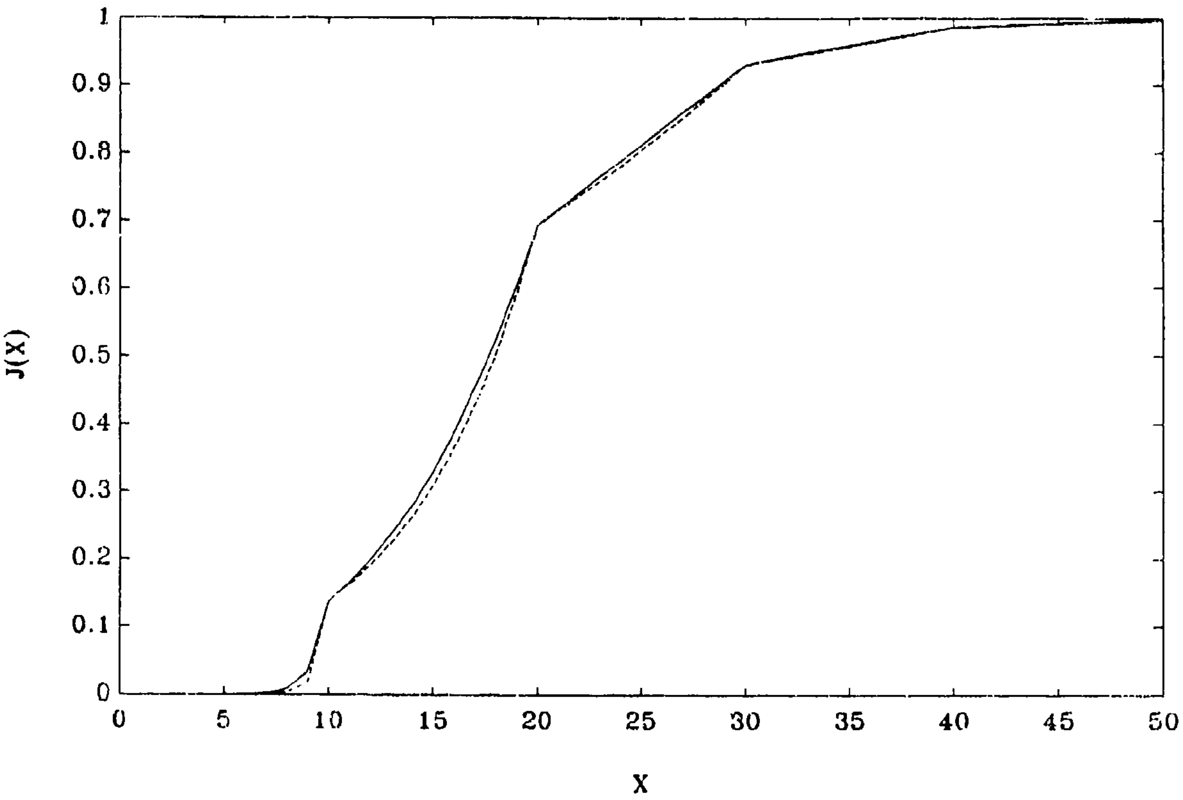


Figure 4.4: Maximum expected surviving value of an asset with $W = 1, n = 10, p_i = 0.8$ for target dependent lethality probabilities (solid line) and target independent lethality probabilities (dashed line).

have plotted the maximum expected surviving value versus the number of weapons assigned to its targets. The dashed curve is for the case of target independent kill probabilities $p_i = 0.8$. The solid curve is for the case of target dependent kill probabilities $\{p_i\} = [.7, .7, .7, .8, .8, .8, .9, .9, .9, .9]$. In this case we chose the kill probabilities so that $\hat{J}(5n) \approx J(5n)$. Note that there are still sections in which the function $\hat{J}$ is convex. However, the overall effect is that the function $\hat{J}$ is "almost concave".

Let us now consider the case in which the kill probability of a weapon-target pair is independent of the target but the lethality probability of a target-asset pair is dependent on the target. Repeating a similar argument to the above, we again find that for the case of target dependent lethality probabilities, the function $J_k(X)$ is smoother than that for the case of target independent lethality probabilities. This is illustrated in figure 4.4 where we have considered the problem of a single asset with $W = 1, n = 10$ and $p_i = 0.9$. We have plotted the maximum expected surviving value versus the number of weapons assigned to its targets. The dashed curve is for the case of target independent lethality probabilities $\pi_i = 0.8$. The solid curve is for the case of target dependent lethality probabilities $\{\pi_i\} = [.8, .8, .8, .9, .9, .9, 1, 1, 1, 1]$. In this case we chose the lethality probabilities so that $\hat{J}(5n) \approx J(5n)$. Note that the effect of target dependent lethality probabilities is again to smooth the function. However, in this case the effect is less pronounced than for the case of target dependent kill probabilities.

Figure 4.5 illustrates the combined effect of having target dependent kill and lethality probabilities. We considered the problem of a single asset with $W = 1$ and $n = 10$. The dashed curve is for the case of target independent parameters, $p_i = 0.8, \pi_i = 0.9$. The solid curve is for the case of target dependent kill probabilities, $\{p_i\} = [.7, .7, .7, .8, .8, .8, .9, .9, .9, .9]$, and target dependent lethality probabilities $\{\pi_i\} = [.8, .8, .8, .9, .9, .9, 1, 1, 1, 1]$. One can see that unlike the dashed curve, the solid curve is almost concave. For all practical purposes, the solid curve is concave in the region of a heavy defense. This implies that, in the case of target dependent parameters, the addition of a single additional weapon to the defense of an asset always has a significant effect. In the case of target independent parameters, if the number of weapons assigned to the defense of an asset is a multiple of $n$ then the addition of a few more weapons has a negligible effect on the expected surviving value of the asset. This means that, unless $n$ more weapons can be added to the defense of the asset, no more weapons should be added. In some situations it will not always be possible to assign a multiple of $n$ number of weapons to the defense of an asset, and so it is advantageous to have target dependent param .ters (if feasible).

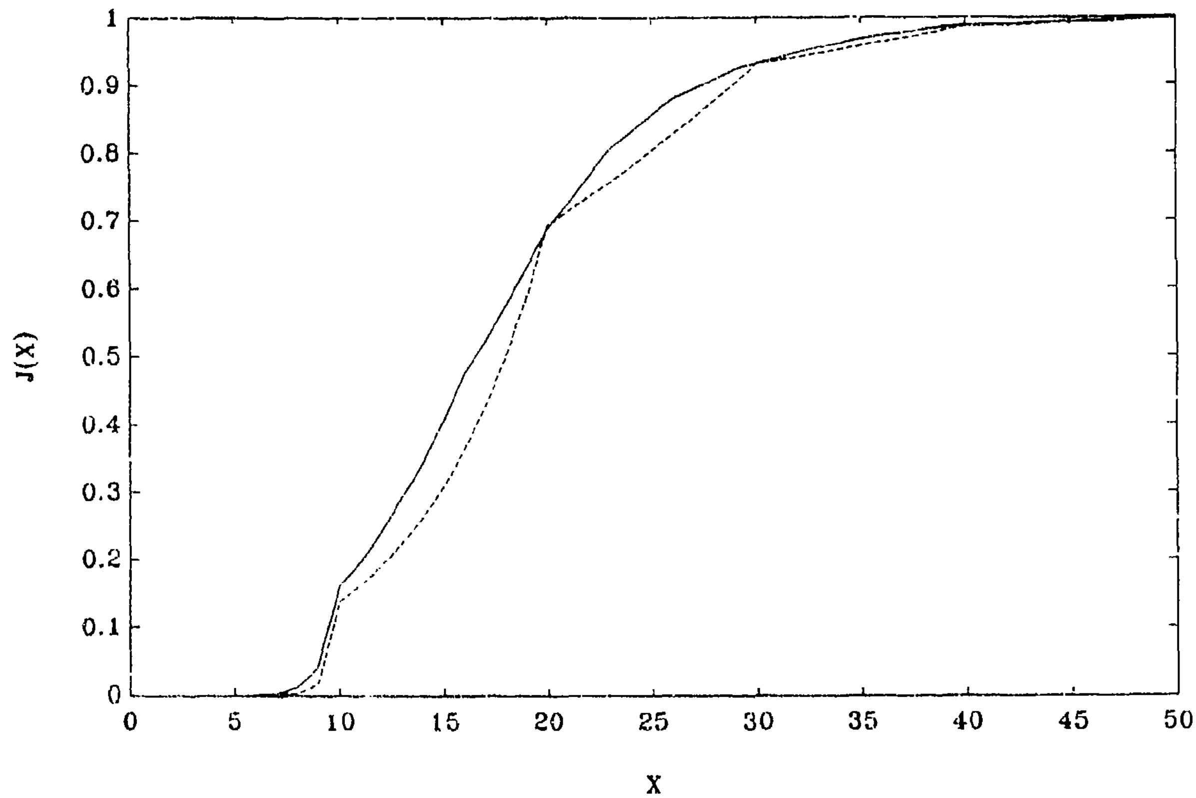


Figure 4.5: Maximum expected surviving value of a unit valued asset versus the number of weapons assigned to its targets for (a) target dependent parameters (solid curve) and (b) target independent parameters (dashed curve).

### 4.3.2 Solution of the Main Problem

Let us define $\bar{J}_k$ to be the concave hull of the function $J_k$ (defined in problem 4.6). We will approximate problem 4.7 by its concave hull. This approximate problem can then be solved to obtain a sub-optimal solution.

---

**Problem 4.8** *The approximate problem (APR) to problem 4.7 is given by:*

$$\max_{\vec{X} \in Z_+^K} \bar{J}(\vec{X}) = \sum_{k=1}^{K} \bar{J}_k(X_k)$$

$$\text{subject to} \qquad \sum_{k=1}^{K} X_k = M.$$

---

Since $\bar{J}$ is a separable concave function then theorem B.1 can be applied to show that the MMR algorithm produces the optimal solution. Let $\vec{\bar{X}}^*$ denote the optimal solution of the approximate problem 4.8. By the nature of the MMR algorithm, we can show that for all but one of the assets

$$J_k(\bar{X}_k^*) = \bar{J}_k(\bar{X}_k^*).$$

Let the asset for which this equality does not hold be asset $v$. Also let $\vec{X}^*$ denote the optimal solution to the true problem 4.7. Using the same analysis as in the previous section we can then show that

$$J(\vec{\bar{X}}^*) + \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*) \geq J(\vec{X}^*) \geq J(\vec{\bar{X}}^*). \tag{4.10}$$

Therefore the optimal solution to the approximate problem can be used to obtain upper and lower bounds on the optimal value of true problem 4.7.

Notice that the solution to the approximate problem 4.8 is a near optimal solution to the true problem 4.7. The difference in value of these two solutions is bounded by:

$$J(\vec{X}^*) - J(\vec{\bar{X}}^*) \leq \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*). \tag{4.11}$$

If $\bar{J}(\bar{X}_v^*) = J(\bar{X}_v^*)$ then $\vec{\bar{X}}^*$ is optimal for 4.7. Also note that if $\bar{J}(\bar{X}_v^*) \neq J(\bar{X}_v^*)$, then by slightly increasing or slightly decreasing the number of weapons one can obtain a problem for which the solution to the approximate problem 4.8 is also optimal for the true problem 4.7. We now state our result as a theorem:

**Theorem 4.3** *Consider the Static Asset-Based problem in which the kill probability of a weapon-target pair depends solely on the target. Let $\vec{\bar{X}}^*$ be the optimal solution to the approximate problem defined in 4.8 obtained by the use of the MMR algorithm. Let $\vec{X}^*$ denote the optimal solution of the true problem 4.7 then*

$$J(\vec{\bar{X}}^*) + \bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*) \geq J(\vec{X}^*) \geq J(\vec{\bar{X}}^*).$$

*Furthermore if we let*

$$\varepsilon = \max_k \max_{\{0 \leq X_k \leq \ell_k\}} \bar{J}_k(X_k) - J_k(X_k)$$

*then*

$$J(\vec{X}^*) - J(\vec{\bar{X}}^*) \leq \varepsilon.$$

*Proof:* The first part of the theorem has already been proved. The second part is obtained by upper bounding the difference $\bar{J}(\bar{X}_v^*) - J(\bar{X}_v^*)$ by its maximum possible value. ■

Therefore we can obtain a suboptimal solution $\vec{\bar{X}}^*$ to the problem as well as an upper bound on the optimal value. Furthermore, if the number of weapons is slightly increased or slightly decreased then the algorithm produces the optimal solution for the corresponding problem.

## 4.4 Approximation of the Asset-Based Problem by a Target-Based One

In this section we will provide a heuristic for solving the Asset-Based problem. In this heuristic, the objective function of the Asset-Based problem is lower bounded by a concave function. If this lower bound is used as the objective function, then it can be shown that the resulting problem is equivalent to a Target-Based problem. Algorithms for the Target-Based problem can then be used to solve the approximate problem.

The approximation that will be used is good in the case of a "strong defense" (i.e. a defense in which all assets are defended and, for each asset, the expected number of targets which survive the weapon engagements is much less than one). For such a case the survival probability of each of the targets will be small. Furthermore, all of the assets will be defended. In the case of a weak defense the approximation will be good for the assets that are defended, but bad for the assets that are left undefended. We will see that, because of this, the algorithm will perform badly for problems which require a preferential defense strategy.

Recall that the objective function for the Asset-Based problem is given by

$$J = \sum_{k=1}^{K} W_k \prod_{i \in G_k} (1 - \pi_i \prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}}).$$

If an asset is defended in the optimal strategy, then we know that sufficient weapons will be assigned to its targets so as to make the probability that one or more of them survives small. If this is the case, then for those targets we can assume that

$$\prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}} << 1. \tag{4.12}$$

We can now use a binomial expansion of $J$ to conclude that

$$J \approx \sum_{k=1}^{K} W_k [1 - \sum_{i \in G_k} \pi_i \prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}}].$$

Note that this approximation is only valid if the inequality 4.12 holds for all assets. This means that all assets must be defended and the defense of each must be strong enough so that the inequality 4.12 holds for all targets. Therefore for many practical problems this approximation may not be appropriate. For each target $i \in G_k$, let its value be that of the associated asset, i.e. $V_i = W_k$. The approximation to $J$ can be written as

$$J \approx \sum_{k=1}^{K} W_k - \sum_{i=1}^{N} V_i \pi_i \prod_{j=1}^{M} (1 - p_{ij})^{x_{ij}}.$$

Therefore, it is easy to see that, the Asset-Based problem with this approximation as the objective is equivalent to a Target-Based problem with the same targets and kill probabilities and with target values given by $V_i\pi_i$. In other words, the value of a target is the expected destroyed value of the asset to which it is aimed if the target is not engaged, but all other targets aimed for that asset are destroyed. Therefore, targets which can potentially do a lot of damage are assigned large values.

Note that this approximation is a lower bound to the true objective function whereas the approximation that was made in section 4.3 (i.e the concave hull approximation) was an upper bound. Let us illustrate this approximation by a simple example. Consider the case of 10 identical assets each of unit value and each having 10 targets aimed at it. Let us assume a single kill probability for all weapon-target pairs and a single lethality probability for all targets. We will use the values: $p = 0.8, \pi = 0.9, n = 10$ and $W = 1$. Note that in the optimal strategy for the approximate problem the weapons assigned to the defense of an asset are spread as evenly as possible among the targets. This is because all targets will be assigned the same value $V_i = 0.9$. This is also the optimal strategy for the true problem. The decision variables are therefore the number of weapons that must be assigned to the defense of each of the assets. This decision will depend on the shape of the function $J(X)$ which is the expected surviving value of the asset if $X$ weapons are assigned to its defense. Let $J(X)$ denote this function for the true objective and let $\tilde{J}(X)$ denote the approximation. In figure 4.6 we have plotted the true function $J(X)$ as well as the lower bound approximation $\tilde{J}(X)$. Note that the approximation is only good in the region $X > 20$. Therefore the approximation is poor for the assets that are not defended.

This approach of approximating the Asset-Based problem by a Target-Based one has the advantage that methods that have already been developed for the Target-Based problem can be used to solve it. However, this particular approach has a serious disadvantage. Consider again the problem above. Let us assume that the defense has 100 weapons. The optimal solution[3] to this problem is to defend 5 of the assets with 20 weapons each. This results

[3]The optimal solution to this problem can be obtained with the algorithm described in section 4.3

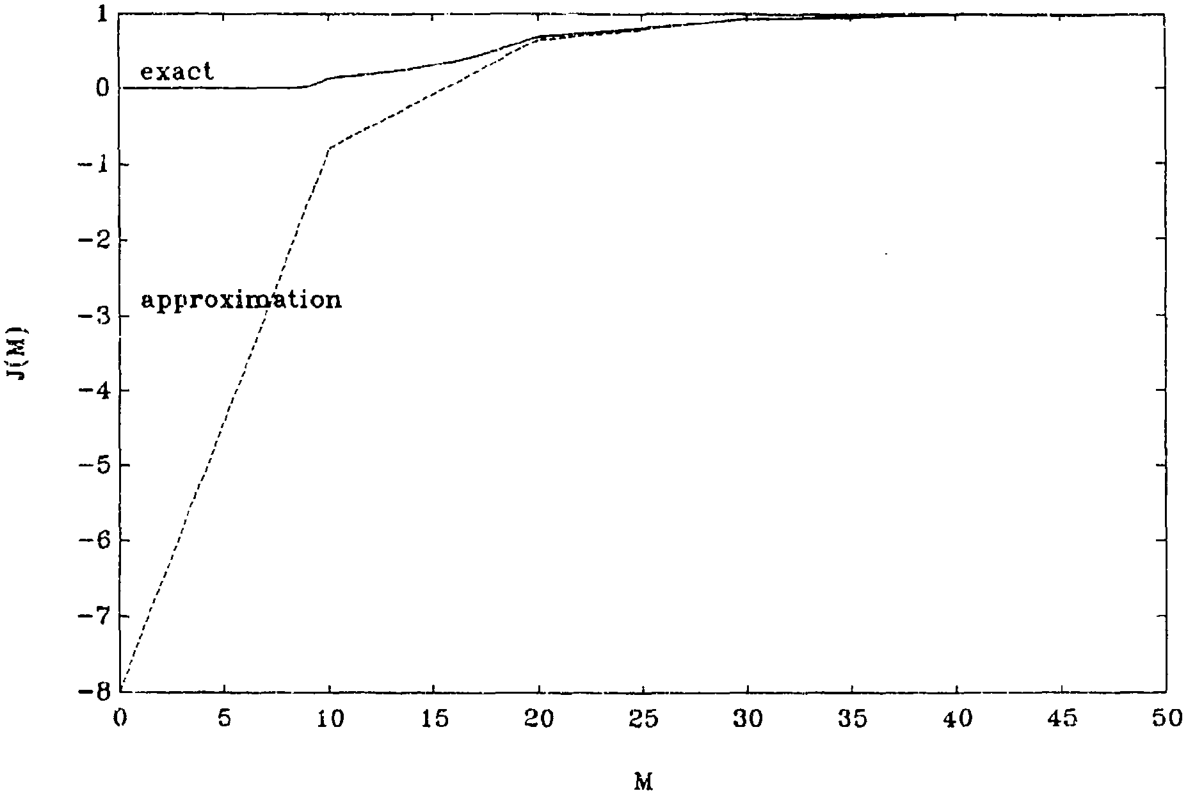


Figure 4.6: The maximum expected surviving value of an asset (solid line) vs. the number of weapons assigned to its defense for the case $W = 1, n = 10, p = 0.8, \pi = 0.9$. The dashed line is the lower bound approximation.

in an optimal value of 3.465. On the other hand the optimal solution for the approximate problem is to assign a single weapon to each of the targets. This results in an expected value of 1.374. Therefore the value of the solution for the approximate problem is roughly 40% of that of the optimal value. We can therefore conclude that the method described in this section may perform badly on certain problems. In particular the algorithm performs poorly on problems which require a preferential defense strategy.

The analysis of this section was performed to illustrate the fact that "reasonable" heuristics may misbehave when applied on certain problems. This fact supports our belief that one should not design algorithms based solely on intuition. Algorithms should be supported by analytical results. The aim of our research is to provide these analytical results which can then be used as a basis for heuristics.

## 4.5 Sensitivity Analysis

In this section we will present some sensitivity analysis results. These results will help us decide the importance of the role of each of the parameters in the optimization problem. This information will be useful in determining how accurately each of the parameters should be measured.

### 4.5.1 Optimal Value Sensitivity Analysis

We will present sensitivity analysis results in this subsection for the case of a single kill probability and a single lethality probability. The following baseline problem will be used:

<u>Baseline Problem Definition</u>

Number of weapons: $M = 200$,

Number of targets: $N = 100$,

Number of assets: $K = 10$,

Number of targets aimed at each asset: $n_k = 10, \qquad k = 1, \ldots, 10$,

Value of each asset: $W_k = 1, \qquad k = 1, \ldots, 10$,

Kill probability of each weapon-target pair: $p = 0.8$, Lethality probability of each target: $\pi = 1$.

We will vary the parameters $p, \pi, M$ and $n_k$ individually and see how the optimal value of the problem changes. As we vary the kill probability $p$ we will denote the optimal value by $J(p)$. Similar notation will be used for the other parameters. Since we do not have an algorithm that guarantees optimal solutions for the problem, we will compute upper and lower bounds on the optimal value. The algorithm presented in section 4.2 will be used to compute a solution to the problem as well as an upper bound on the optimal value. The expected value of the sub-optimal solution will be plotted with a solid line. The upper bound will be plotted with a dashed line. The plot for the optimal value will lie between these plots. Note that for some of the plots the algorithm produces the optimal solution. In these cases no dashed curve will be visible.

In figure 4.7 we have four plots. In plot (a) the upper and lower bounds on the optimal value is plotted versus the kill probability $p$ which is the same for all weapon-target pairs. Note that the dashed curve is almost identical to the solid curve. This means that the solution produced by the algorithm is almost optimal for all values of the kill probability. Also note that for the values of interest to us ($0.5 \leq p \leq 0.9$) the optimal value is very sensitive to the kill probability. Small increases in the kill probability can result in large increases in the optimal value.

In plot (b) the optimal value is plotted versus the lethality probability $\pi$. Note that there is no dashed curve because the solution was optimal. Here we find that the optimal value decreases almost linearly with the lethality probability. It therefore appears that the lethality probability does not play an important role in the optimization problem.

In plot(c) we have plotted the optimal value versus the number of weapons for $M = 100, 150, 200, 250, 300$. For these values of $M$ the algorithm produced the optimal solution. We find that the optimal value increases almost linearly with the number of weapons.

In plot(d) the upper and lower bounds on the optimal value is plotted versus the number of targets aimed for each of the assets. We kept the weapon-target ratio fixed at 2:1. Again note that the algorithm is optimal for most of the plot. Here we find that the plot appears

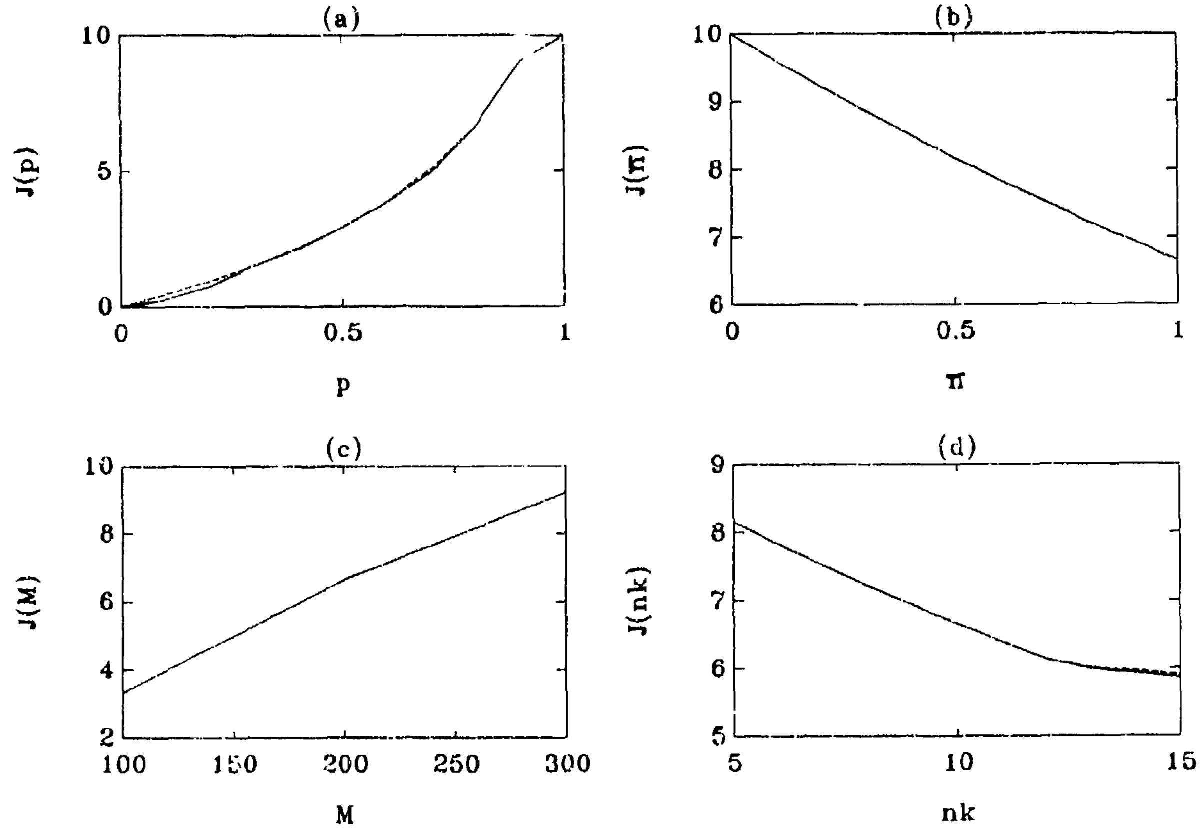


Figure 4.7: Upper and lower bounds on the optimal value for the baseline problem, as a function of (a) the kill probability (b) the lethality probability (c) the number of weapons and (d) the number of targets aimed for an asset (with $M = 2N$).

to be that of a piecewise-linear convex function. We also find that as the number of targets aimed for each asset increases, the optimal value decreases. This implies that, if the number of assets is kept fixed then as the size of the attack increases (i.e $n_k$ increases for each $k$) the defense's arsenal must be increased at a *greater* rate to maintain the same level of performance. This gives the offense a tremendous advantage because, if we keep the kill and lethality probabilities fixed, then a small increase in the offense's arsenal has to be countered by a larger increase in the defense's arsenal if the defense wishes to maintain the same level of performance.

### 4.5.2 Optimal Solution Sensitivity Analysis

In the previous subsection we considered what happens to the optimal value as various problem parameters were varied. In this section we will see what happens to the optimal solution of the problem as each of the parameters is varied. We will first present some analytical results and then some numerical results.

Let us assume that all weapon-target pairs have a kill probability of $p$ and that all targets have a lethality probability of $\pi$. We will also assume that the number of targets $n$, aimed for an asset, is the same for all assets. Finally we will assume that all assets have a value of unity. Since we do not have an optimal algorithm for the problem we will consider the solution produced by the sub-optimal algorithm presented in section 4.3. Because of the nature of the algorithm and the uniformity of the problem being studied, the solution it produces will have the property that all but one of the defended assets will be defended by the same number of weapons. The number of weapons assigned to the other defended asset will be less than or equal to the number of weapons assigned to the others. Because of this property, a more convenient way to state the solution is in terms of the number of assets defended. Given this information one may then compute the optimal number of weapons to be assigned to each of the targets.

We will compute an approximate value for the number of assets defended in the solution of the algorithm. For $k \in Z_+$, let $J(kn)$ denote the expected surviving value of an asset if

$k$ weapons asre assigned to each of the targets aimed for it. Note that since all assets are identical then this function is the same for each. We have

$$J(kn) = (1 - \pi(1-p)^k)^n.$$

Define $\ell \in Z_+$ as follows:

$$\frac{J(\ell n)}{\ell} = \max_k \frac{J(kn)}{k}.$$

In other words the tangent to $J(nk)$ through the origin touches the function at the point $k = \ell$. Note that, because all assets are identical, the value of $\ell$ is the same for all. The MMR algorithm will assign $n\ell$ weapons to the defense of each of the defended assets but one. The remaining weapons will go to the other defended asset. Therefore the number of defended assets, which we will denote by $d$, is given by:

$$d = \frac{M}{n\ell}.$$

If $d$ is not integral then $\lfloor d \rfloor$ assets will be defended with $\ell n$ weapons each and another asset will be partly defended with the remaining weapons. Recall that the function $J(kn)$ is at first convex and then becomes concave. If we relax the constraint the $k$ is integral then we can find the point at which $J$ changes from a convex to a concave function by fincding the point at which its second derivative changes sign (from positive to negative). Setting the second derivative to zero we find that this change occurs at the point $\hat{k}$ given by

$$\hat{k} = \frac{-\log(n\pi)}{\log(1-p)}.$$

In other words the value of $k$, $\hat{k}$, at which this occurs satisfies the equation

$$n\pi(1-p)^{\hat{k}} = 1.$$

Note that $n\pi(1-p)^{\hat{k}}$ is the expected number of surviving targets if $\hat{k}$ weapons are assigned to each of the targets. This implies that if the expected number of surviving targets is greater than one then the objective function is convex otherwise it is concave. Note that, in general, $\hat{k}$ is non-integral so we need to say that $J(nk)$ is convex if $k \leq \lfloor \hat{k} \rfloor$ and it is concave if $k \geq \lceil \hat{k} \rceil$. Therefore a good approximation for the value of $\ell$ is $\lceil \hat{k} \rceil$. Using this approximation we have

$$d \approx \frac{M}{n\lceil \frac{-\log(n\pi)}{\log(1-p)} \rceil}. \tag{4.13}$$

Note that if $d \geq K$ then all of the assets are defended. Let us consider an example. Consider the following problem: $M = 200, N = 100, K = 10, V = 1, n = 10, p = 0.7, \pi = 0.8$. The solution obtained by the sub-optimal algorithm is to defend all assets with 2 weapons per target. Using our approximation we find $d \approx 10$; so that in this case the approximation is good. Note that $d$ varies linearly with $M$, it varies roughly inversely with $N$, it increases as $\pi$ decreases, and it increases as $p$ increases. All of the results are as we would expect. This approximation provides us with a simple estimate of the optimal strategy of any given problem.

We will next examine how the solution of the algorithm presented in section 4.2 varies with different parameter values. Note that in subsection 4.5.1 we considered the sensitivity of the optimal *value* to changes in the parameter values. Here we are considering the sensitivity of the optimal ***solution***. We will use the same baseline problem that was used in subsection 4.5.1. Because of the symmetry of the problem, the solution can be completely characterized by the number of defended assets. Note that in the solution to the problem, the same number of weapons is used to defend each of the defended assets except one. The number of weapons assigned to this special asset is less than the number assigned to each of the others. We will include this special asset, as a fraction, in the number of assets defended. This fraction is the ratio of the number of weapons assigned to defend the asset and the number of weapons assigned to each of the other defended assets. In figure 4.8 we have plotted the number of defended assets vs each of the parameters $p, \pi, M$ and $n$.

In plot (a) of figure 4.8 we have plotted the number of defended assets versus the kill probability. Note that small changes in the kill probability can result in significant changes in the strategy. Plot (b) contains the plot for the lethality probability. Here we find that changes in $\pi$ do not affect the optimal strategy. This suggests that the lethality probability plays a small role in the optimization problem. Plot (c) contains the plot for the number

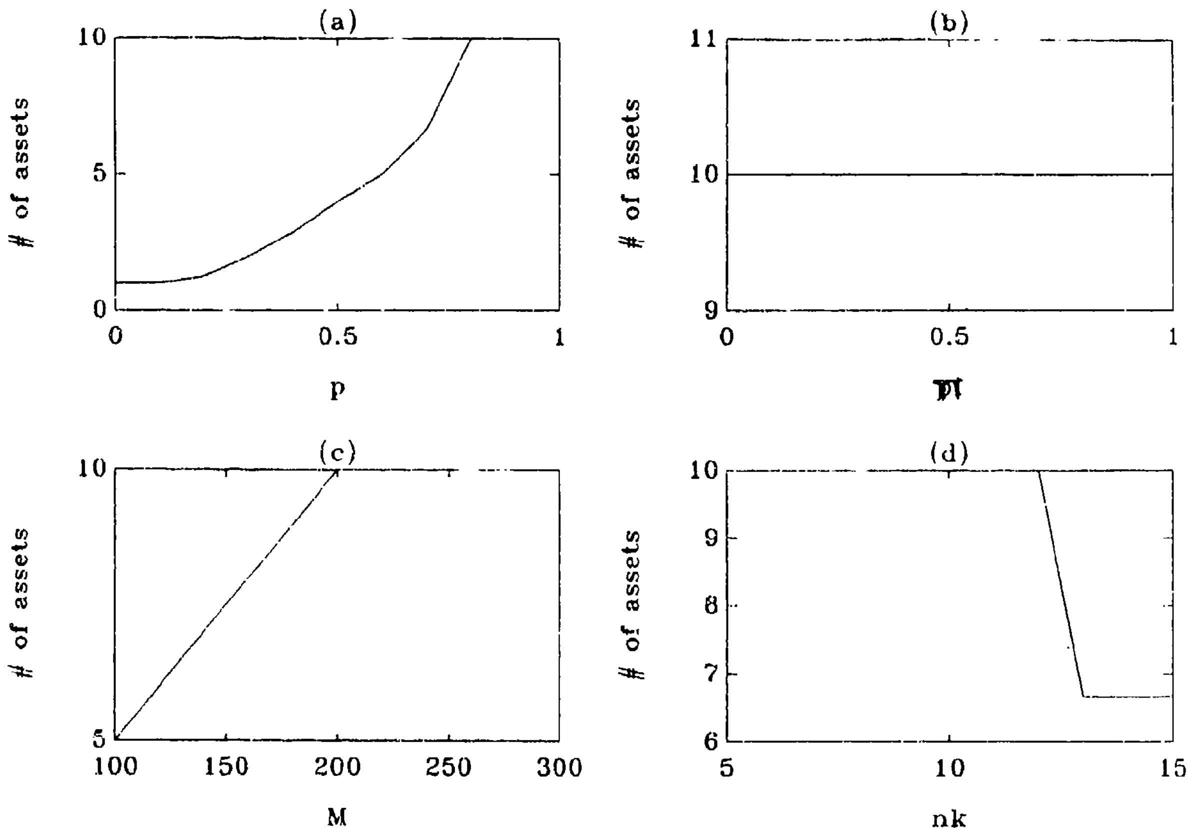


Figure 4.8: Number of defended assets as a function of (a) the kill probability (b) the lethality probability (c) the number of weapons and (d) the number of targets aimed for each asset (with $M = 2N$).

of weapons. As the number of weapons increases more assets are defended until they are all defended. Plot (d) contains the plot for the number of targets per asset with a fixed 2:1 weapon to target ratio. Note the sudden change in the defense strategy as $n_k$ changes from 12 to 13. For the case $n_k = 12$ the defense assigns two weapons per target to defend its assets. However, for the case $n_k = 13$, two weapons per target is not enough so it has to start using three weapons per target for 6 of the assets and two weapons per target for one of the assets (which will be included as a fraction of 2/3). Therefore the defense only defends $6\frac{2}{3}$ assets.

### 4.5.3 Asset Value Sensitivity Analysis

In this thesis we do not plan to address the question of how the various parameters ($p_{ij}, W_k$ etc.) of the problem are obtained. However, in deciding values for these parameters one should have an idea of the ranges within which they should lie. For example, if the spread of the asset values is small then the solution of the resulting problem might be the same as the solution of the problem in which all asset values are equal. If such is the case, then either all assets can be considered as having the same value or one must increase the spread in the values to reflect the fact that some assets are of greater value than others. If, on the other hand, the spread of the asset values is large, then the solution of the resulting problem will be such that no weapons are assigned to the assets of low value. If such is the case, then either the low valued assets could be removed from the problem to decrease its size, or if one really wants to consider these low valued assets as a part of the problem, then the spread in the values should be decreased. This example suggests that there is a range within which the values should be assigned if the resulting solution is to be meaningful. In this section we will compute such a range for the asset values for a simple problem.

We will investigate the sensitivity of the optimal assignment of a static Asset-Based problem to changes in the asset values. We will consider the case of two assets under the assumptions that each is attacked by $n$ targets and that the kill probability of each weapon-target pair is $p$ and that the lethality probability of each target is 1. We will also assume that the value of one of the assets is unity while that of the other is $W$ with $W \geq 1$. This last assumption does not restrict the class of problems considered since the values of the assets can always be scaled so that the smaller valued asset has a value of unity. We are interested in the value of $W$ above which only one asset is defended (the asset of value $W$) and below which both of the assets are defended. We will assume that the number of weapons is a multiple of $n$ (i.e. $M = \kappa n$).

The optimal assignment for this problem has the property that the optimal number of weapons assigned to the defense of each of the assets is a multiple of $n$. This can be shown as follows. Between multiples of $n$ the expected surviving value of an asset is convex (see section 4.2). Therefore, for all but one of the assets, the number of weapons assigned to the defense of the asset must be a multiple of $n$. Since $M$ is also a multiple of $n$ then the number of weapons assigned to the other defended asset must also be a multiple of $n$. Therefore the property holds. Let this multiple be $\kappa_1$ for the higher valued asset and $\kappa_2$ for the other. Since $W \geq 1$ the possible optimal values of the pair $(\kappa_1, \kappa_2)$ are $(\kappa, 0), (\kappa - 1, 1), \ldots, (\lceil \kappa/2 \rceil, \lfloor \kappa/2 \rfloor)$. Let $W_i$ denote the value of $W$ at which the solution $(\kappa, 0)$ changes to the solution $(\kappa - i, i)$. We have,

$$W_i = \frac{(1-(1-p)^i)^n}{(1-(1-p)^\kappa)^n - (1-(1-p)^{\kappa-i})^n}.$$

The value of $W$ at which the defense's strategy changes from defense of one asset to the defense of both assets will be the maximum, over $i$, of $W_i$.

$$W^* = \max_{0<i<\lfloor \kappa/2 \rfloor} W_i. \tag{4.14}$$

For values of $p$ close to unity, and/or for small values of $n$, a reasonable approximation that can be made is the following:

$$(1-(1-p)^{\kappa-i})^n \approx 1 - n(1-p)^{\kappa-i} \qquad \forall \quad 0 \leq i \leq \lfloor \kappa/2 \rfloor.$$

Using these approximations we have that:

$$W_i \approx \frac{(1-(1-p)^i)^{n-1}(1-p)^i}{n(1-p)^\kappa}$$

Let us take $i$ to be a continuous variable and set the derivative of $\bar{W}_i$, with respect to $i$, to zero. This leads to the equation $n(1-p)^i = 1$. One can show that this is the only stationary point and that the function is concave in the region around this stationary point. This leads to the conclusion that the value of $i$ satisfying this equation is the point at which the function is maximum. Substituting back into the equation we find that

$$W^* \approx \frac{(1-\frac{1}{n})^{n-1}}{n^2(1-p)^\kappa}$$

If we make the approximation $(1-1/n)^n \approx e^{-1}$ then we have

$$W^* \approx \frac{1}{en^2(1-p)^\kappa}$$

If we assume that $\kappa$ is even and that the weapons are divided evenly between the two assets then the expected number of targets which survive in each asset is given by $\mathcal{L} = n(1-p)^{\frac{\kappa}{2}}$. We can write $W^* \approx (e\mathcal{L}^2)^{-1}$. Therefore as the expected number of surviving targets decreases, the ratio of the asset values above which a single asset is defended, increases. Also note that if $\mathcal{L} > \frac{1}{\sqrt{e}} \approx 0.6$ that it is always optimal to assign weapons only to the higher valued asset.

Table 4.1 contains the exact values of $W^*$ for the case of $M = 4n$ (ie a 2:1 weapon to target ratio) for various values of $p$ and $n$. Table 4.2 contains the values obtained by using the approximation $[en^2(1-p)^4]^{-1}$.

The results in tables 4.1 and 4.2 suggest that our approximation to $W^*$ is a good one for the problems which are of interest to us. If this is the case then it implies that the factor which determines the range of the asset values is the expected number of surviving targets if the defense attempts to save both assets. The expected number of surviving targets is called the *target leakage* in the literature. Therefore the number of assets defended should be such that the resulting target leakage is sufficiently small.

## 4.6 Concluding Remarks

In this chapter we presented the Static Asset Based WTA problem as well as a sub-optimal algorithm for solving it under the assumption of target dependent kill and lethality probabil-

| | $p$=0.6 | $p$=0.7 | $p$=0.8 | $p$=0.9 |
|---|---|---|---|---|
| $n = 2$ | 4.9 | 13.2 | 50.3 | 450 |
| $n = 4$ | 1.2 | 3.4 | 16.2 | 182 |
| $n = 6$ | 1.0 | 1.4 | 6.9 | 99 |
| $n = 8$ | 1.0 | 1.0 | 3.4 | 60 |
| $n = 10$ | 1.0 | 1.0 | 2.1 | 39 |
| $n = 12$ | 1.0 | 1.0 | 1.6 | 26 |
| $n = 14$ | 1.0 | 1.0 | 1.3 | 18 |

Table 4.1: The value of $W$ above which only one asset is defended. $M = 4n$

| | $p$=0.6 | $p$=0.7 | $p$=0.8 | $p$=0.9 |
|---|---|---|---|---|
| $n = 2$ | 3.6 | 11.3 | 57.5 | 920 |
| $n = 4$ | 1.0 | 2.8 | 14.4 | 230 |
| $n = 6$ | 1.0 | 1.3 | 6.4 | 102 |
| $n = 8$ | 1.0 | 1.0 | 3.6 | 57 |
| $n = 10$ | 1.0 | 1.0 | 2.3 | 37 |
| $n = 12$ | 1.0 | 1.0 | 1.6 | 25 |
| $n = 14$ | 1.0 | 1.0 | 1.2 | 19 |

Table 4.2: The value of $W$ above which only one asset is defended using the approximation $[en^2(1-p)^4]^{-1}$.

ities. Computational experimentation suggests that the solution produced by this algorithm is either optimal or near optimal for most problems. We also presented some sensitivity analysis results which will prove helpful in choosing parameter values for the problem.

The main conclusions that can be drawn from the results of this chapter are as follows:

- The general problem is difficult because it is a more general version of the Static Target-Based problem which has been shown to be NP-Complete [2].

- In the case of a single class of weapons the algorithm that we have proposed (in section 4.3) provides near-optimal solutions. We conjecture that, if this approach is used as a heuristic for the case of multiple weapon classes then the resulting solution will also be near-optimal.

- The optimal value and optimal solution of the problem is quite sensitive to changes in the kill probability, but appears to be insensitive to changes in the lethality probability.

- If the number of assets, and the kill and lethality probabilities are kept fixed then, as the number of offense weapons increases, the number of defensive weapons must increase at a greater rate if the defense wishes to maintain the same level of performance.

The efficient solution of static Asset-Based problems will require the use of parallel coputers. Therefore, it will be necessary to investigate parallel algorithms for solving the problem. Also the information and computers needed to solve the problem may be geographically dispersed. This suggests the use of distributed algorithms for the problem. Some preliminary work in this area can be found in [18] and [19].

# Chapter 5

# The Dynamic Asset-Based Problem

In this chapter we will consider the dynamic version of the Asset-Based WTA problem. Recall that the Target-Based WTA problem, discussed in chapter 3, is a special case of the Asset-Based Problem in which a single target is directed at each asset. Also all static problems are special cases of the corresponding dynamic problems. Therefore the three previously studied problems, the Static Target-Based WTA problem, the Dynamic Target-Based WTA problem and the Static Asset-Based WTA problem are all special cases of the Dynamic Asset-Based WTA problem. Hence, this problem is the most difficult and complex because of its generality. However, it is also the most important because of its generality. Simplifying assumptions are needed to reduce the complexity so as to derive efficient suboptimal algorithms.

We will make the assumption that the kill probability of a weapon-target pair and the lethality probability of a target-asset pair depend solely on the asset to which the target is directed. Under these assumptions the number of decision variables per stage equals the number of assets. Without these assumptions the number of decision variables per stage equals the product of the number of available weapons and the number of surviving targets. Therefore, the assumptions greatly reduce the dimensionality of the problem.

These are restrictive assumptions which will be violated in most practical problems. If the assumptions do not hold the solution method described in this chapter can still be applied after some modifications have been made. There would, however, be a degradation in the performance of the method. This degradation would vary with the degree by which the assumptions are violated. Furthermore, in a practical situation, if each defensive weapon cannot engage all offensive targets the dynamic strategy will lose some of its performance advantage over the static strategy. Therefore, although we will conclude that the performance advantage of the dynamic strategy is roughly twice that of the static strategy, in practice this performance advantage will be less.

In section 5.1 we will define the general problem and discuss its complexity. In section 5.2 we will give a mathematical statement of the problem under the assumptions of asset dependent kill and lethality probabilities. Because of the extreme complexity of the problem, we will only consider the case of two stages. We will show that, under the assumptions made, the decision variables are the number of weapons to be used in the first stage and the optimal assignment of these weapons. In section 5.3 we will discuss the problem of finding the optimal number of weapons to be used in stage 1. We will find that this is a difficult problem because of the presence of multiple local maxima. In section 5.4 we will assume that the optimal number of weapons to be used in the first stage is known and discuss the problem of finding the optimal assignment of these weapons. We will present a sub-optimal algorithm for this problem. In section 5.5 we will provide a heuristic based on approximating the Asset-Based objective function by a Target-Based one. This approximate problem can then be solved by the methods presented in chapter 3. We will find that the main shortcoming of such an approach is that it cannot produce a truly preferential defense strategy. In section 5.6 we will present several numerical results. We will find that, in general, a dynamic strategy outperforms a static one by a factor of two. Finally in section 5.7 we will make some concluding remarks.

## 5.1 Problem Definition

As in the case of the Dynamic Target-Based problem, this problem consists of a number of time stages. In each stage the results (survival or destruction of each target) of the engagements of the previous stage are observed. Based on these observations, a subset of the remaining weapons is chosen and assigned to the surviving targets. The results of the engagements of this assignment is then observed and the process is repeated. Hence we are dealing with a "shoot-look-shoot-..." strategy. The objective is to choose and assign weapons at each stage so as to maximize the total expected value of the surviving assets at the end of the final stage of the engagement. Note that the problem will be re-solved after each stage because the results of that stage can be observed. This means that one is only interested in obtaining assignments for the present stage. By the principle of optimality, it is implicitly assumed that optimal assignments will be used in all subsequent stages.

We will first define the general problem. In the next section we will consider the special case of two stages under the assumptions that the kill probability of a weapon-target pair depends solely on the asset to which the target is directed, and the lethality probability of a target-asset pair depends solely on the asset. The following notation will be used. The definitions of all additional notation can be found in Appendix A.

$K \stackrel{\text{def}}{=}$ the number of defense assets,

$T \stackrel{\text{def}}{=}$ the number of time stages,

$N \stackrel{\text{def}}{=}$ the initial number of targets,

$M \stackrel{\text{def}}{=}$ the total number of weapons,

$W_k \stackrel{\text{def}}{=}$ the value of asset $k$, $\quad k = 1, 2, \ldots, K$,

$G_k \stackrel{\text{def}}{=}$ the set of targets aimed for asset $k$ initially, $k = 1, 2, \ldots, K$,

$n_k(t) \stackrel{\text{def}}{=}$ the number of targets aimed for asset $k$ in stage $t$, $k = 1, 2, \ldots, K$,

$p_{ij}(t) \stackrel{\text{def}}{=}$ the probability that weapon $j$ destroys target $i$ in stage $t$ if assigned to it, $i = 1, 1 \ldots, N$, $\quad j = 1, 2, \ldots, M$,

$\pi_i \stackrel{\text{def}}{=}$ the lethality probability of target $i$ on the asset to which it is aimed, $i = 1, 2, \ldots, N$.

The decision variables will be denoted by:

$$x_{ij} = \begin{cases} 1 & \text{if weapon } j \text{ is assigned to target } i \text{ in stage } 1 \\ 0 & \text{otherwise} \end{cases}$$

Note that we only need to solve for the decision variables in stage 1. The decision variables for all subsequaent stages will be obtained after the outcome of the weapon-target engagements of the previous stage is observed.

The *target state* of the system at the end of the first stage will be defined as the set of surviving targets. This state will be denoted by an $N$- dimensional binary vector $\vec{u} \in \{0,1\}^N$ and represented by

$$u_i = \begin{cases} 1 & \text{if target } i \text{ survives stage 1} \\ 0 & \text{if target } i \text{ is destroyed in stage 1.} \end{cases}$$

The *weapon state* of the system at the end of stage one will be defined as the set of available weapons. This state will be denoted by an $M$-dimensional binary vector $\vec{w} \in \{0,1\}^M$ and represented by:

$$w_j = \begin{cases} 1 & \text{if weapon } j \text{ was not used in stage 1,} \\ 0 & \text{if weapon } j \text{ was used in stage 1.} \end{cases}$$

The target state evolves stochastically. The stochastic evolution of the target state in stage 1 depends on the assignment decisions made in stage 1. Given a first stage assignment of $\{x_{ij}\}$, the state at the start of the second stage is an $N$-dimensional random vector. The probability that $u_i$ is 1 is the probability that target $i$ survives the first stage. The probability that $u_i$ is 0 is the probability that target $i$ is destroyed in the first stage. The distribution of the random variable $u_i$ is therefore given by:

$$\Pr[u_i = k] = k \prod_{j=1}^{M} (1 - p_{ij}(1))^{x_{ij}} + [1 - k] \left\{ 1 - \prod_{j=1}^{M} (1 - p_{ij}(1))^{x_{ij}} \right\}, \tag{5.1}$$

$$\text{for} \qquad k = 0, 1, \qquad i = 1, 2, \ldots, N.$$

Equation 5.1 will be called the *target state evolution* of the system.

The evolution of the weapon state is deterministic and depends on the assignments made in the first stage. The evolution is given by:

$$w_j = 1 - \sum_{i=1}^{N} x_{ij}, \qquad j = 1, 2, \ldots, M. \tag{5.2}$$

This simply says that weapon $j$ is available in the second stage if and only if it is not used in the first stage. Equation 5.2 will be called the *weapon state evolution* of the system.

We will let $J_2^*(\vec{u}, \vec{w})$ denote the *optimal* value of a $T-1$ stage problem in which the initial target state is $\vec{u}$ and the initial weapon state $\vec{w}$. This problem has the same form as the $T$ stage problem which is being defined. The $T-1$ stage problem can be defined in terms of the optimal values of $T-2$ stage problems etc. The $T-(T-1)$ or single-stage problem can be defined in terms of the optimal values of 0-stage problems. If the target state at the end of the final stage is $\vec{u}$ and the weapon state at the end of the final stage is $\vec{w}$ (which would be $[0,\ldots,0]$ for an optimal strategy) then the optimal value of the 0-stage problem is given by

$$J_T^*(\vec{u}, \vec{w}) = \sum_{k=1}^{K} W_k \prod_{i \in G_k} (1 - \pi_i u_i).$$

In other words this is the value if no more weapons are assigned and, targets which have been destroyed have a lethality probability of 0 while each target $i$ which survived all stages has a lethality probability of $\pi_i$. We can now state the problem as follows.

---

**Problem 5.1** *The Dynamic Asset-Based problem (DAB) can be stated as:*

$$\min_{\{x_{ij}\}} J_1 = \sum_{\vec{\omega} \in \{0,1\}^N} \Pr[\vec{u} = \vec{\omega}] J_2^*(\vec{\omega}, \vec{w})$$

$$\text{subject to} \quad x_{ij} \in \{0,1\}, \qquad i = 1,2,\ldots,N \quad j = 1,2,\ldots,M,$$

$$\text{with} \quad w_j = 1 - \sum_{i=1}^{N} x_{ij}.$$

---

The objective function is the sum over all possible stage 2 target states of the probability of occurrence of that state times the optimal value given that state. Note that the distribution of the stage 2 target state and the stage 2 weapon state both depend on the first stage assignment. The first constraint restricts each weapon to be assigned at most once in the first stage. The second constraint is due to the weapon state evolution.

This problem is considerably more difficult than the static one. This can be illustrated by attempting to use a straightforward dynamic programming approach to the problem.

Let us consider a two stage problem. The number of possible weapon subsets that can be chosen in the first stage is $2^M$. If $m_1$ weapons are used in stage 1 the number of possible assignments that must be checked is $N^{m_1}$. If $\tilde{N}$ of the $N$ targets are engaged in the first stage the number of possible outcomes is $2^{\tilde{N}}$. If $\tilde{N}$ of the $N$ targets survive stage 1 and $m_2$ weapons are available in stage 2 then the number of assignments that must be checked to obtain the optimal value for this outcome is $\tilde{N}^{m_2}$. These numbers show the enormous number of computations that will be required if a straightforward dynamic programming approach is used. Note that to simply evaluate the expected value of a first stage assignment requires a tremendous computational effort. Besides the problem of dimensionality there is also the difficulty of solving the static problem in the last stage. Several of these static problems must be solved corresponding to the different possible outcomes. Recall that the objective function for the static problem was neither convex nor concave. Since there are no efficient algorithms for obtaining the optimal value of the static problem, we cannot even evaluate an arbitrary assignment for the dynamic problem. These difficulties have forced us to make some simplifying assumptions. We believe that this simplified problem will reflect the overall behaviour of the more general problem.

## 5.2 The Two-Stage Problem with Asset Dependent Kill and Lethality Probabilities

Because of tremendous complexity of the general version of the problem we will make some simplifying assumptions. We will only consider the case of two stages since the complexity of the problem grows exponentially with the number of stages. We will make the assumption that the kill probability of a weapon-target pair depends solely on the asset to which the target is directed. Therefore the kill probability of any weapon on a target aimed for asset $k$ will be denoted by $p_k$. We will also assume that the lethality probability of a target-asset pair depends solely on the asset. Therefore the lethality probability of each of the targets aimed for asset $k$ will be denoted by $\pi_k$.

Because of the assumption of weapon independent kill probabilities, we can let the decision variables be the number of weapons assigned to each target in each stage. Furthermore the assumptions imply that all targets directed to a specific asset are identical. Therefore, in each stage, the number of weapons asigned to any two targets aimed for the same asset cannot differ by more than one. In other words the weapons assigned to defend an asset in a stage must be spread as evenly as possible among the surviving targets aimed for that asset. This result can be used to simplify the problem even further by defining the decision variables as the number of weapons assigned to defend each asset in each stage. We can therefore let the decision variables be $m_1$, the number of weapons to be used in stage one, and $\vec{X} \in Z_+^K$ the assignment of these $m_1$ weapons in stage 1, where $X_k$ represents the number of weapons assigned to defend asset $k$ in stage one. The individual target assignments can be obtained by spreading these weapons as evely as possible among the targets aimed for asset $k$.

Our assumptions can also be used to simplify the representation of the target state. Since all targets directed to a specific asset are identical then we can represent the target state by $\vec{n}(2)$ where $n_k(2)$ is the number of targets aimed for asset $k$ that survive the first stage.

The state $\vec{n}(t)$ of the system evolves stochastically. This evolution depends on the weapon assignments made. Because we assumed that the engagement of a target by a weapon in a stage is independent of all other engagements in all stages then, given an assignment for the first stage, the state $n_k(2)$ of asset $k$ evolves independently of all other assets. The state for each asset evolves as follows. To simplify the expression we have left out the subscript $k$ from the variables $n_k(\cdot), p_k(\cdot), q_k(\cdot)$.

$$\begin{aligned}
&\Pr[n(2) = j \mid X = \chi] = \\
&= \sum_{\ell=\underline{\ell}}^{\bar{\ell}} \binom{\chi - n(1)\lfloor \frac{\chi}{n(1)} \rfloor}{\ell} q(1)^{\ell \lceil \frac{\chi}{n(1)} \rceil} [1 - q(1)^{\lceil \frac{\chi}{n(1)} \rceil}]^{\chi - n(1)\lfloor \frac{\chi}{n(1)} \rfloor - \ell} \times \\
&\quad \times \binom{n(1)\lfloor \frac{\chi}{n(1)} \rfloor + n(1) - \chi}{j - \ell} q(1)^{(j-\ell)\lfloor \frac{\chi}{n(1)} \rfloor} [1 - q(1)^{\lfloor \frac{\chi}{n(1)} \rfloor}]^{n(1)\lfloor \frac{\chi}{n(1)} \rfloor + n(1) + \ell - \chi - j}
\end{aligned} \tag{5.3}$$

for

$$j = 0, 1, \ldots, n(1).$$

where

$$\underline{\ell} = \max\{j + \chi - n(1)(\left\lfloor \frac{\chi}{n(1)} \right\rfloor + 1), 0\} \quad \text{and} \quad \overline{\ell} = \min\{\chi - n(1)\left\lfloor \frac{\chi}{n(1)} \right\rfloor, j\}$$

This evolution can be explained as follows. If $X_k$ is a multiple of $n_k(1)$ then $n_k(2)$ is a binomial random variable with success probability[1] $(1-p_k(1))^{\frac{X_k}{n_k(1)}}$. If $X_k$ is not a multiple of $n_k(1)$, then some targets will be assigned $\lfloor \frac{X_k}{n_k(1)} \rfloor$ weapons while the others will be assigned $\lceil \frac{X_k}{n_k(1)} \rceil$ weapons. The distribution of the random variable $n_k(2)$ is obtained by convolving two binomial distributions. The success probability of one of these distributions is given by $(1-p(1))^{\lfloor \frac{X}{n(1)} \rfloor}$, while the success probability of the other is given by $(1-p(1))^{\lfloor \frac{X}{n(1)} \rfloor}$. The variables $\underline{\ell}$ and $\overline{\ell}$ were introduced to take care of the boundary conditions of the convolution.

Let $J_2^*(\vec{n}(2), M)$ denote the optimal value of the second stage problem with target state $\vec{n}(2)$ and $M$ weapons. Also let $\mathcal{S}$ denote the set of all possible outcomes of the first stage

$$\mathcal{S} = \{\vec{s} \in Z_+^K | s_k \in \{0, 1, \ldots, n_k(1)\}\}.$$

We will state the two-stage problem in terms of the optimal values of single stage problems. The single-stage problem is simply a static problem. However, we can use the same recursive definition that was used for the two-stage problem to define the single-stage problem in terms of optimal values of 0-stage problems. Note that in the optimal strategy no weapons will be available after the second stage. Denote the target state after stage two, the final stage, by $\vec{n}(3)$. The optimal value of the 0-stage problem is given by:

$$J_3^*(\vec{n}(3), 0) = \sum_{k=1}^{K} W_k(1-\pi_k)^{n_k(0)}.$$

[1] See Appendix A for the definition of a binomial random variable as well as the term success probability.

In other words $J_3^*$ is the total expected value of the surviving assets if the target state is $\vec{n}(3)$ and no more weapons are fired.

---

**Problem 5.2** *The Two-Stage, Dynamic, Asset-Based (TDAB) problem with asset dependent kill and lethality probabilities can be stated as:*

$$\begin{aligned}
\max_{\{\vec{X}\}} J_1 &= \sum_{\vec{s}\in S} \Pr[\vec{n}(2) = \vec{s}] J_2(\vec{s}, M - m_1) \\
\text{subject to} \quad & X_k \in Z_+, \qquad k = 1, \ldots, K \\
\text{and} \quad & \sum_{k=1}^{K} X_k = m_1.
\end{aligned}$$

---

One can see that even the statement of the problem is a formidable task even under the assumption that the kill and lethality probabilities are solely asset dependent.

By the principle of optimality, the assignments used in the second stage must be optimal. Therefore, the only decision variables over which the objective function is to be optimized are $m_1$ and $\vec{X}$, which is the number of weapons to be used in the first stage $m_1$ and the assignment of these weapons to assets $\vec{X}$. We will therefore denote the optimal value for the case in which $m_1$ weapons are used in the first stage with assignment $\vec{X}$ by $J_1(m_1, \vec{X})$.

---

**Problem 5.3** *The Dynamic Asset-Based problem may also be stated as:*

$$\begin{aligned}
&\max_{m_1 \in Z_+} \left\{ \max_{\vec{X}\in Z_+^K} J_1(m_1, \vec{X}) \right\} \\
&\text{subject to} \quad \sum_{k=1}^{K} X_k = m_1, \\
&\text{and} \quad 0 \le m_1 \le M.
\end{aligned}$$

---

If we fix $m_1$ then the inner subproblem can be written as

**Problem 5.4** *(Assignment subproblem):*

$$\max_{\{\vec{X}\in Z_+^K\}} J_1(m_1, \vec{X})$$

$$\text{subject to} \qquad \sum_{k=1}^{K} X_k = m_1.$$

If we can solve the assignment subproblem, then the original problem can be solved as follows. Let $\vec{X}^*$ denote the optimal assignment of the subproblem 5.4. Note that this optimal assignment depends on the value of $m_1$. However, this value is implicit in the solution since $\sum_{k=1}^{K} X_k^* = m_1$. The solution to the original problem may now be obtained by solving the following:

**Problem 5.5** *(Main problem):*

$$\max_{m_1\in Z_+} J_1(m_1, \vec{X}^*)$$

$$\text{subject to} \qquad 0 \le m_1 \le M.$$

Each of the problems 5.4 and 5.5 will be considered separately. Our efforts will be concentrated on the solution of problem 5.4 since we will show that problem 5.5 has many maxima and hence, in general, a global search will have to be done to obtain the optimal solution.

## 5.3 Optimal Number of First-Stage Weapons

Let us assume that we can solve the assignment subproblem 5.4 and consider the problem 5.5 for the case of $T = 2$. Recall that, for the Target-Based case, the corresponding problem had multiple minima. Since the Target-Based problem is a special case of the Asset-Based problem then one can conclude that problem 5.3 will also have multiple local maxima. Consider, for example, the case $M = 14, K = 3, \vec{\pi} = [1,1,1]$ and $p_k(t) = 0.9$. (Note that this is the Asset-Based version of the problem that was used to illustrate that, for the Target-Based problem, the expected value as a function of the number of weapons used in

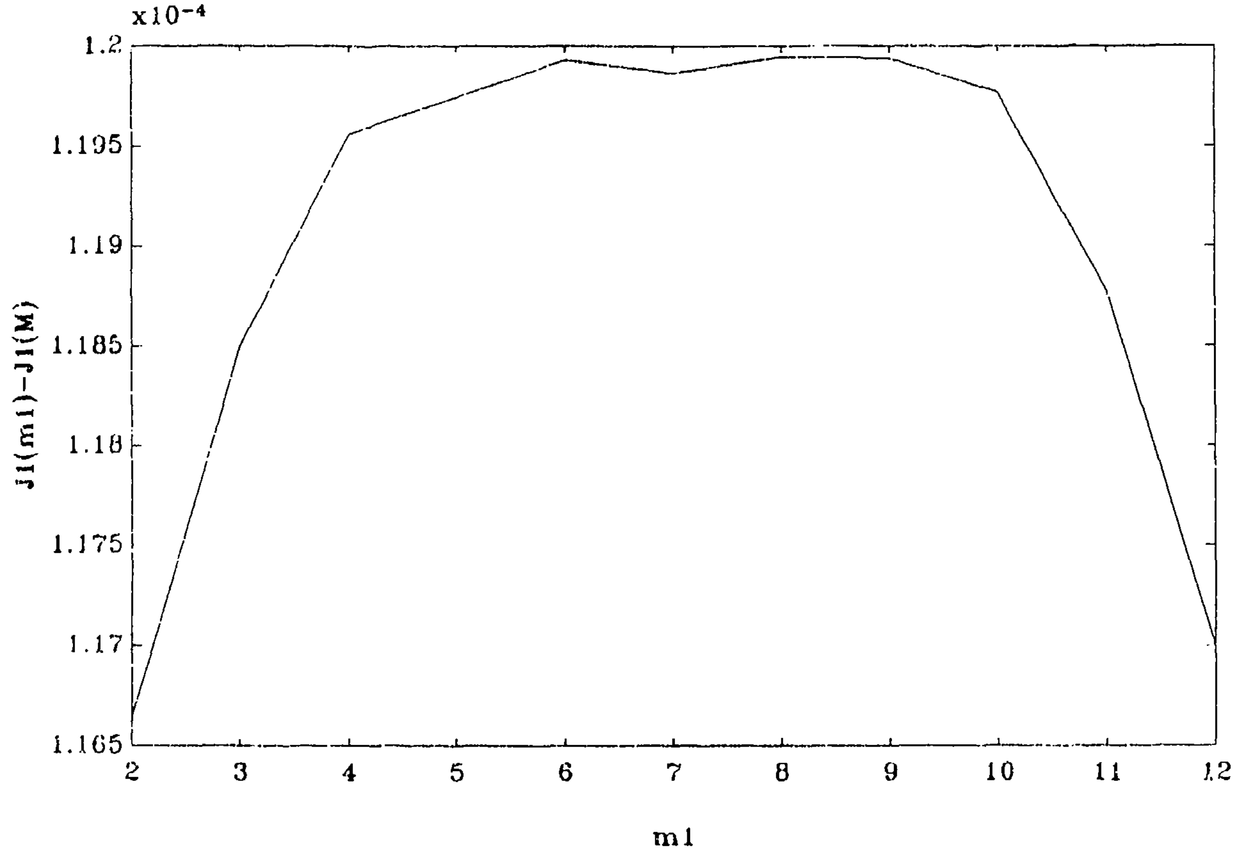


Figure 5.1: An example of the two-stage dynamic asset-based problem for which multiple maxima exists. Plot of the expected two-stage value $J_1(m_1)$ minus the static value $J_1(M)$ vs. the number of weapons used in stage 1, $m_1$, with $M = 14, K = 3, \vec{n}(1) = [1,1,1], p = 0.9$.

the first stage had multiple minima). In figure 5.1 we have plotted $J_1(m_1, \vec{X}^*(m_1)) - J_1(M)$ versus $m_1$ for this problem. The optimal value of the static strategy was subtracted from that of the dynamic strategy to obtain a scale on which the different maxima are visible. Furthermore, we have only plotted the cases $m_1 = [2, \ldots, 12]$. Again this was done so that the different maxima will be visible. The difference in value of the local maxima is so small that for all practical purposes the solution for any of them will be satisfactory.

Therefore, to obtain the global maxima one must essentially do a global search. For most practical purposes however, a local maximum will suffice. To obtain a local maximum a simple local search algorithm (like the one presented in section 3.4 for the Target-Based version of the problem) can be used. If several processors are available then the local search algorithm can be run on each of them simultaneously with different initial solutions. The best local maximum may then be taken.

Problem 5.3 is an important one since it is used to determine the optimal number of weapons to use in the present stage. However, it is also a difficult problem to solve because the objective function is not unimodal. Our belief is that in practice any local maxima will suffice since we conjecture that the difference in the values of any two local maxima will be negligible compared to the value of any one of them. The reason why all solutions cannot be checked is because of the computational requirements for evaluating each solution. This computation can be reduced by making good approximations. One approach that has been used [10] is to approximate the marginal return of increasing the value of $m_1$ by one and decreasing the value of $m_2$ by one. If this is positive then $m_1$ is increased and the process is repeated. Similarly the marginal return of decreasing the value of $m_1$ by one and increasing the value of $m_2$ by one can be approximated etc.

## 5.4 Optimal Assignment of the First-Stage Weapons

In this section we will consider the assignment subproblem 5.4. In this problem the number of weapons to be used in the first stage is fixed and the objective is to assign these weapons optimally. Note that for the static version of this problem we were able to obtain a sub-optimal algorithm but not an optimal one. In this section we will provide a suboptimal algorithm as well. This algorithm is similar to that used to solve the static problem in that it approximates the objective function by a concave, separable one. We will illustrate the algorithm for the case of two stages.

Since there are only two stages then $m_2 = M - m_1$. Let $J_1(\vec{X})$ denote the expected value for a first stage assignment of $\vec{X}$, (with $\sum_{k=1}^{K} X_k \leq m_1$), and $m_2$ weapons are assigned in the second stage optimally. The function $J_1(\vec{X})$ is non-separable (with respect to the assets) and non-concave We will approximate this function by a function $\tilde{J}_1(\vec{X})$ which is both separable and concave.

Let $e_k$ denote the $k^{\text{th}}$ column of the $K$-dimensional identity matrix and let $X_k$ be a non-negative integer. Consider the one-dimensional function $J_1(X_k e_k)$. This is the expected value if, in the first stage, $X_k$ weapons are assigned to asset $k$ and no other weapons are assigned in this stage while in the second stage $m_2$ weapons are assigned optimally. An example of this function is given in figure 5.2 (the solid line). For this example we used $K = 2, k = 1, \vec{n}(1) = [10, 10], \vec{W} = [1, 1]$, and $\vec{p}(t) = [.4, .4]$. The number of weapons used in stage 2 was fixed at 20.

Note that, as a function of multiples of $n$, the function is convex and then becomes concave. This property was observed for the static problem as well. However note that, between multiples of $n$ the function is convex for small $X$ and concave for larger values of $X$. This is unlike the static case for which the function was always convex between multiples of $n$. The reason for this is that, even if only a subset of the targets aimed for an asset are engaged, there is still a significant increase in value because the remaining targets will be engaged in the second stage. We have also included the concave hull (the dashed line) of the function in the plot. We will denote the concave hull of this function by $\tilde{J}_1(X_k e_k)$. Note that the concave hull is a very good approximation to the function.

Let us denote the $K$-dimensional zero vector by $\vec{0}$. The approximation to the function

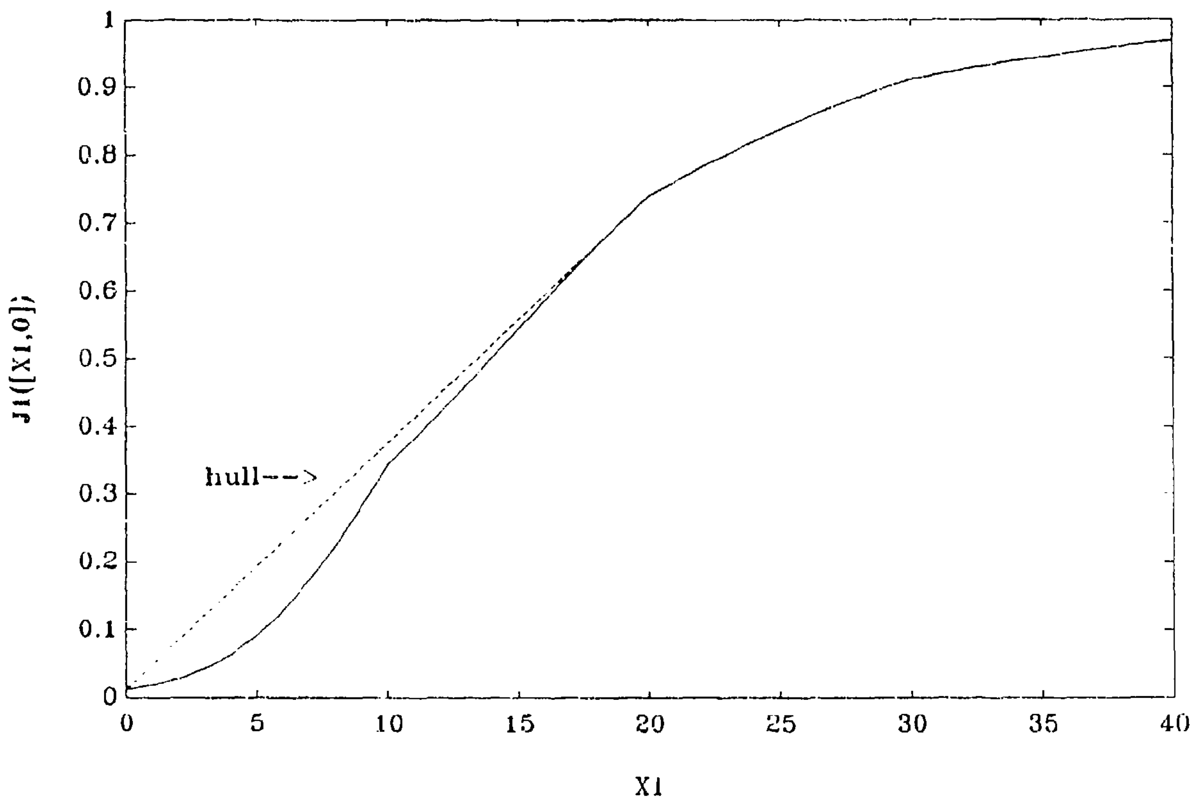


Figure 5.2: An example the expected two-stage value, $J_1(\vec{X})$, plotted along a coordinate direction for a two-asset problem.

$J_1(\vec{X})$ which we will use is given by:

$$\tilde{J}_1(\vec{X}) \equiv J_1(\vec{0}) + \sum_{k=1}^{K}[\tilde{J}_1(X_k e_k) - J_1(\vec{0})] \tag{5.4}$$

where the function $\tilde{J}_1(X_k e_k)$ is the concave hull of the function $J_1(\vec{X})$ along the $k^{\text{th}}$ coordinate direction. Along the coordinate directions this approximation is the concave hull as given in figure 5.2. The values for the interior points are obtained by summing the increases along each coordinate and adding the value at the origin. Note that the value at the origin is the optimal value of the corresponding static problem with $m_2$ weapons since no weapons are used in stage 1.

Note that the function $\tilde{J}_1(\vec{X})$ is concave and separable with respect to the assets. Furthermore, note that if $m_1 = M$, (i.e. all weapons are used in the first stage) then the problem is a static one and the approximation used is the same as the approximation that was used in the suboptimal algorithm for the static problem that was presented in chapter 4. Also note that if only enough weapons are used in stage 1 to defend one of the assets, then the approximation is the same as the exact function because along the coordinate directions through the origin both functions are equal in the region in which the asset is defended. Therefore, in the limits of small and large values of $m_1$ the approximation is good.

In figure 5.3 we have plotted the function $J_1(\vec{X})$ versus $X_1$ and $X_2$ for the example used for figure 5.2. In figure 5.4 we have plotted the corresponding approximation $\tilde{J}_1(\vec{X})$. We only evaluated the functions at points where $X_1$ and $X_2$ were multiples of $n$. Note that the approximation is good if $X_1 \geq 20$ and $X_2 = 0$ or if $X_2 \geq 20$ and $X_1 = 0$. This is where the solution will lie if only one of the assets is defended. The approximation is also good in the region $X_1 \geq 20, X_2 \geq 20$. This is where the solution will lie if both assets are defended. Also note that the approximation is an upper bound on the true function. The algorithm is given in figure 5.5.

The suboptimal solution is obtained by solving the problem with the approximate function as the objective. The value of this solution is then evaluated using the exact function.

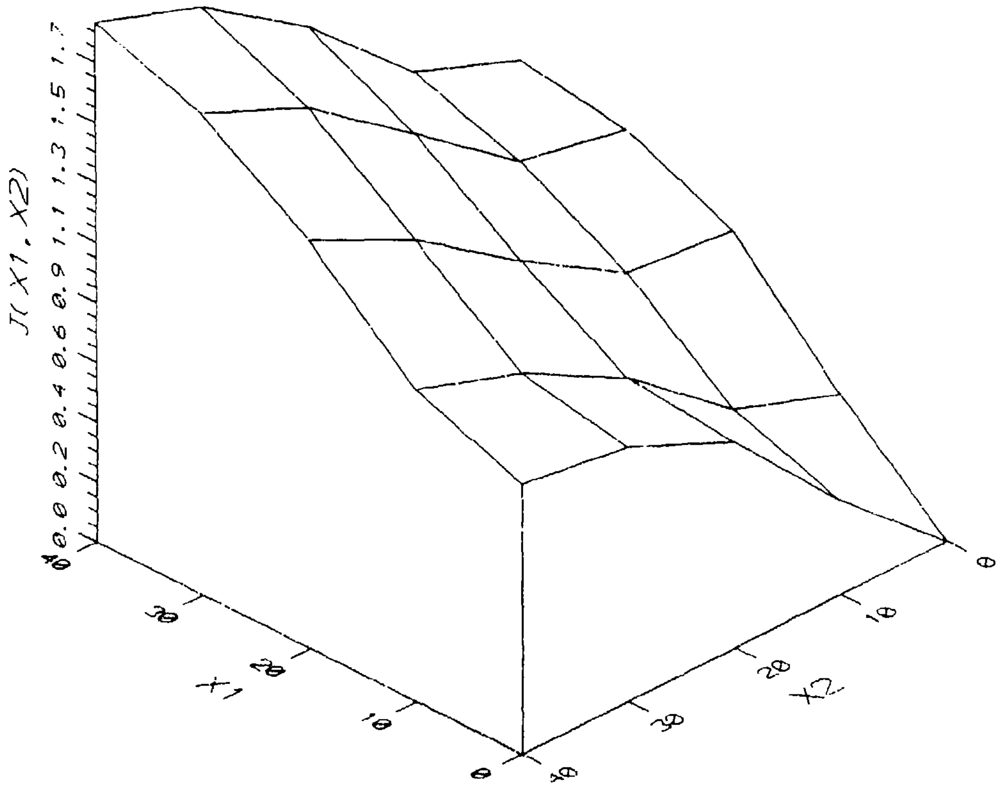


Figure 5.3: The expected two-stage value $J_1(\vec{X})$, if $X_k$ weapons are assigned to defend asset $k$ in stage 1 and 20 weapons are reserved for stage 2 with $K = 2, n_k(1) = 10, W_k = 1, p_k(t) = .4, \pi_k = 1$ for $k = 1, 2$.

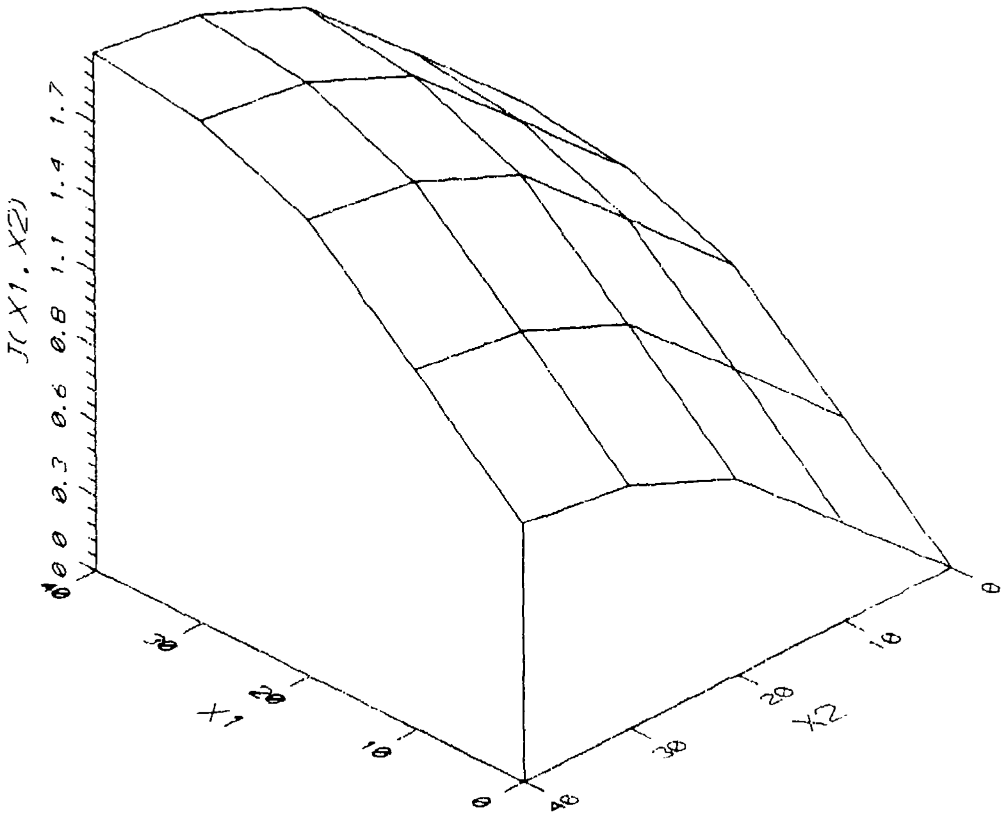


Figure 5.4: The concave hull approximation, $\tilde{J}_1(\vec{X})$, of the function $J_1(\vec{X})$ given in figure 5.3.

**procedure** DAB
**begin**

Pick a value for $m_1$;
Compute the approximate function $\tilde{J}_1(\vec{X})$;
Use MMR algorithm to assign the $m_1$ first stage weapons using
$\tilde{J}_1(\vec{X})$ as the objective function;
This assignment will be the sub-optimal solution for the dynamic problem;
Evaluate value of assignment using simulations;

**end**

Figure 5.5: Algorithm for the Dynamic Asset-Based problem

However, since the approximate function is an upper bound then if we evaluate the solution using the approximate function then we can obtain an upper bound on the optimal value of the problem.

**Theorem 5.1** *The function $\tilde{J}_1(\vec{X})$ defined by equation 5.4 is an upper bound to the function $J_1(\vec{X})$, i.e*

$$\tilde{J}_1(\vec{X}) \geq J_1(\vec{X} \qquad \text{for} \quad \vec{X} \in Z_+^K.$$

*Proof:* Let us first show that the marginal return of adding weapons to an asset *decreases* as the number of weapons assigned to the other assets increases. Let us first consider the case in which weapons are being added to the defense of two assets $j$ and $k$. We want to show that:

$$J_1(\vec{X} + \chi_k e_k) - J_1(\vec{X}) \geq J_1(\vec{X} + \chi_k e_k + \chi_j e_j) - J_1(\vec{X} + \chi_j e_j).$$

where $\chi_k$ and $\chi_j$ are positive integers. Let $E(r_j, r_k)$ denote the expected value given that $r_j$ of the targets aimed for asset $j$ survive stage 1 and that $r_k$ of the targets aimed for asset $k$ survive stage 1. We then have:

$$\begin{aligned} &J_1(\vec{X} + \chi_k e_k) - J_1(\vec{X}) - J_1(\vec{X} + \chi_k e_k + \chi_j e_j) + J_1(\vec{X} + \chi_j e_j) = \\ &= \sum_{r_j=0}^{n_j(1)} \sum_{r_k=0}^{n_k(1)} \Pr(n_j(1) = r_j) \Pr(n_k = r_k)[E(n_j, r_k) + E(r_j, n_k) - E(n_j, n_k) - E(r_j, r_k)]. \end{aligned} \tag{5.5}$$

Consider the difference

$$E(n_j, r_k) + E(r_j, n_k) - E(n_j, n_k) - E(r_j, r_k). \tag{5.6}$$

If $r_j = n_j$ or $r_k = n_k$, then this difference is zero. Let us therefore assume that $r_j < n_j$ and $r_k < n_k$. The functi n $E$ is the expected value of a static problem. Consider any outcome $\vec{n}(2)$ of stage 1 and consider a new problem in which each asset $i$ is duplicated and the duplicate $i'$ has the same number of targets aimed for it. This is done for all assets except assets $j$ and $k$. For asset $j$ we will assume that the original is being attacked by $n_j$ targets while the duplicate $j'$ is being attacked by $r_j$ targets. For asset $k$ we will assume that the original is being attacked by $n_k$ targets while the duplicate $k'$ is being attacked by $r_k$ targets. Let $S$ denote the set of the original assets except $j$ and $k$ and let $S'$ denote the set of the duplicate assets except $j'$ and $k'$. The number of weapons for this problem will also be doubled to $2m_2$. Consider the optimal solution of this new problem. If assets $j$ and $k$ are defended, then the number of weapons assigned to them will be greater than the number assigned to $j'$ and $k'$. On the other hand if they are not defended then the number assigned to $j'$ and $k'$ will be greater than those assigned to $j$ and $k$ (which is zero in this case). Therefore if we solve the problem under the restriction that $m_2$ weapons are to be assigned to the assets $S \cup \{j,k\}$ and $m_2$ weapons are to be assigned to the assets $S' \cup \{j',k'\}$, then the resulting optimal value will be less than or equal to the optimal value of the problem under the restriction that $m_2$ weapons are assigned to the assets $S \cup \{j',k\}$ and $m_2$ weapons are assigned to assets $S' \cup \{j,k'\}$. This is because in the latter problem weapons can be optimally divided between assets $j'$ and $k$ and between assets $j$ and $k'$. The optimal value of the former problem is $E(n_j,n_k) + E(r_j,r_k)$, while the optimal value of the latter problem is $E(n_j,r_k) + E(r_j,n_k)$. Therefore, the difference 5.6 is non-negative which implies that the quantity in 5.5 is non-negative.

This argument can be repeated for any subset of assets to conclude that:

$$J_1(\vec{X}) - J_1(\vec{0}) \leq \sum_{k=1}^{K}[J_1(X_k e_k) - J_1(\vec{0})] \leq \sum_{k=1}^{K}[\tilde{J}_1(X_k e_k) - J_1(\vec{0})] = \tilde{J}_1(\vec{X}) - J_1(\vec{0}).$$

Therefore the approximation is an upper bound to the true objective function. ∎

Note that evaluation of any feasible assignment in stage 1 requires an optimal algorithm to compute the optimal stage 2 value for each possible outcome of stage 1. Since we do not have an optimal static algorithm we can only compute a lower bound on the expected value of the solution of the dynamic algorithm. This is done by using the value of the solution produced by the algorithm described in chapter 4 for the solution of the static problem in stage 2. There is also the problem that the number of possible outcomes is enormous. To overcome this problem we use Monte Carlo simulations. We simulate the first stage outcome and then compute the value given that outcome. Several of the simulations are run and the sample mean is taken as an approximation of the value. These simulations will be discussed in detail in subsection 5.5.1.

An upper bound on the optimal value is obtained as follows. Solve the problem in which the objective (dynamic case) function is replaced by the approximate function $\tilde{J}$. We also need to use an upper bound for the value in the second stage. This can be obtained from the sub-optimal algorithm that was presented for the static problem.

## 5.5 A Dynamic Target-Based Approximation to the Asset-Based problem

In this section we will present a heuristic for the Dynamic Asset-Based problem in which the Asset-Based objective is approximated by a Target-Based one. In section 4.4 we presented a method for approximating the objective function of the Static Dynamic-Based problem with a Target-Based objective function. The resulting Target-Based problem could then be solved using the methods of chapter 3. This provided a sub-optimal solution for the Static Asset-Based problem. If we use this approximation in the final stage of the Dynamic Asset-Based problem then the resulting problem is a Dynamic Target-Based problem. The methods of chapter 3 can then be used to solve the approximate problem. Such a method has been used by Castañon et al [10].

This approach has the advantage that the solution methods for the Target-Based problem can be used in its solution. The approximate problem is also simpler and requires less computation than the true (i.e. the Dynamic Asset-Based) problem. There are, however, two serious limitations to such an approach. These limitations may lead to poor performance of the algorithm in certain situations. These situations are often present in practical problems. We would like to stress that the two limitations to be presented below apply only if a *pure* Target-Based approximation method is used. Modifications can be made to the Target-Based approximation approach to remedy these limitations. Such modifications[2]

[2] The specific details of these modifications were not available to the author.

have been used in the heuristics used by Castañon et al. Therefore, the limitations to be described do *not* apply to their methods. The purpose of presenting the Target-Based approach and its limitations (if used in its pure form) is to emphasize the fact that one must be careful in choosing appropriate approximations. Approximations which may seem "reasonable" may have serious flaws.

The first limitation of the approach is that it cannot produce a truly preferential defense when such a defense is preferable. This limitation will appear in problems in which there are several low kill probability weapons (i.e. problems in which a preferential defense strategy is optimal). The reason for this is as follows. Let us first consider the static Asset-Based problem. Recall that for this problem, the expected surviving value of an asset as a function of the number of weapons assigned to defend the asset is a non-linear, monotonically increasing function. This function is convex if the number of assigned weapons is small and concave if the number of assigned weapons is large. The shape of this function is what is responsible for preferential defense strategies as we will next explain. If few weapons are assigned to defend an asset then the marginal return of adding a weapon is larger than it was for the previously added weapon because the expected surviving value of the asset is convex in this region. Therefore, it is advantageous to continue adding weapons since the marginal return continues increasing. Eventually the number of weapons assigned would be such that the expected surviving value of the asset is concave. Henceforth, it is not advantageous to add weapons since the marginal return is small. This implies that several weapons should be assigned to the assets which are defended and no weapons should be assigned to the other (undefended) assets. If a Target- Based approximation is made the resulting objective function will be concave, even when a small number of weapons are assigned to defend an asset. Therfore, if such an approximation is made, preferential defense strategies will not be optimal. Let us now consider the dynamic version of the asset-based problem. In order to have preferential defense strategies in the first stage the objective function must have the property that it is convex if few weapons have been assigned. Note that the approximation used in section 5.3 was chosen so that this property is maintained.

However, if a Target-Based approximation is made for the last stage of the problem then the objective function in the first stage of the resulting approximate problem will not have the property that for small numbers of assigned weapons the objective is convex. Therefore, if a Target-Based approximation is used for the last stage then it is not possible to obtain a preferential defense in the first stage. However, we will find in the next section that in some situations it is preferable to use a preferential defense strategy even in the first stage of the dynamic problem. We will find that the algorithm described in section 5.3 produces a first- stage preferential defense strategy for such cases. However, if the Target- Based approximation is used then a preferential first-stage defense strategy cannot be obtained. The basic reason for this difference is that the method described in section 5.3 approximates the first-stage objective function and maintains the important characteristics of the true function. On the other hand the Target-Based approximation approach approximates the expected value for the final stage. Therefore the characteristics of this approximation are carried over into the first-stage objective function.

The second limitation of the Target-Based approximation method occurs in problems in which there are few weapons. Note that this is the opposite of the case considered in the previous paragraph (i.e. many weapons). Consider for example the case of two stages with problem parameter values given by $M = 100, N = 100, n_k = 10, K = 10$ $p_k = 0.8, \pi_k = 1, W_k = 1$ for $k = 1, \ldots, K$. We find that for the approximate Target-Based problem all targets will have a value of unity. Furthermore, since the number of weapons is equal to the number of targets then we know that a static strategy will have the same performance as a static one. Therefore, an optimal strategy for the Target-Based approximation problem is to use all 100 weapons in stage 1 and to assign one weapon per target. The value of this assignment is 1.07. Note that this value is even worse than the optimal value for the Static Asset-Based problem which is 3.32. In the next section we will show that the solution produced by the method described in section 5.3, is to use 70 weapons in stage 1 and to defend 7 of the assets with these weapons. The remaining 30 weapons are then used in the second stage. The value of this solution is 6.53. Therefore we find that the Target-Based approximation method can produce very poor results. In fact for this problem the value of the solution produced by the Target-Based approximation method is even worse than the optimal value for the Static problem. We again can conclude that for certain problems the *pure* Target-Based approximation method may perform poorly. On the other hand, the algorithm described in the previous section (section 5.3) can easily handle the problems for which the Target-Based approximation method performs poorly.

Our conclusion is that,in its pure form (i.e. use a Target-Based approximation for the expected value in the final stage and solve the resulting Dynamic Target-Based problem to obtain a suboptimal solution to the Dynamic Asset- Based problem), the Target-Based approximation approach will not perform as well as the method described in section 5.3. However, this approach has certain advantages such as simplicity and the fact that Target-Based algorithms can be used in its solution. For practical problems these advantages may be more important than the advantage of better performance which is the main advantage of the method of the section 5.3. Modifications to the Target-Based approximation approach can be made to remedy these limitations. These modifications will depend on the nature of the problems being solved.

## 5.6 Numerical Results

In this section we will present several computational results for the Dynamic Asset-Based WTA problem. We will use the algorithm given in figure 5.5 to solve the problem and will also provide an upper bound on the optimal value for the problem. Some sensitivity analysis results will also be presented.

The following problem will be used as our baseline problem. We will consider the case of two time stages. The kill probability of each weapon-target pair in each of the stages is $p_k(t) = 0.6$. There are $K = 10$ assets to each of which is aimed $n_k = 10$ targets. The defense has $M = 200$ weapons to intercept these 100 targets. The lethality probability $\pi_k$ of each target is unity. The value $W_k$ of each of the assets is unity. This problem was chosen as the baseline problem because it illustrates the following. The optimal static strategy of this problem is to defend 5 of the 10 assets. However, we will find that in a dynamic scenario it is better to defend nine of the assets in the first stage. Therefore the number of assets defended is almost doubled if a dynamic strategy is used rather than a static one.

### 5.6.1 Discussion of Simulations

As was mentioned in the previous section, our proposed algorithm produces a sub-optimal solution. In order to compute the expected value for this solution one needs an optimal algorithm for the static problem since for each possible first-stage outcome one must find the optimal value for the corresponding static problem. Since this is not available, we will produce a *lower* bound on the value of this solution. This will be done by using a lower bound on the optimal value for each static problem that must be solved. Another difficulty is the number of possible outcomes that must be examined. For example, suppose that in the baseline problem each of the assets had a different value and that in the first stage a single weapon is assigned to each target. If this is the case then for each asset either 10 of the targets aimed for it may survive or 9, …, or 0. Therefore since there are 10 assets the total number of possible outcomes of stage 1 is $11^{10}$. For each of these outcomes one must calculate the corresponding optimal static value. Such a task is overwhelming. This difficulty is overcome by using Monte Carlo Simulations.

We simulate the first stage of the engagement as follows. Let $\vec{X}$ denote the first stage assignment. Because of the uniformity of the problem, the optimal target assignments can be obtained by spreading the weapons assigned to an asset evenly among the targets aimed for that asset. Let us denote the first stage target assignments by $\vec{x}$. The engagements of the weapons on target $i$ is simulated by flipping a coin. The success probability of the coin is $(1 - p_k)^{x_i}$. If the coin toss is a success then we assume that target $i$ survives the first stage, while if the coin toss is a failure then we assume that target $i$ is destroyed in stage 1. This is repeated for all targets to obtain the target state for the second stage. The expected value of this outcome is then computed (actually only bounds on the expected value can be computed because we do not have an optimal algorithm for the static problem). Several

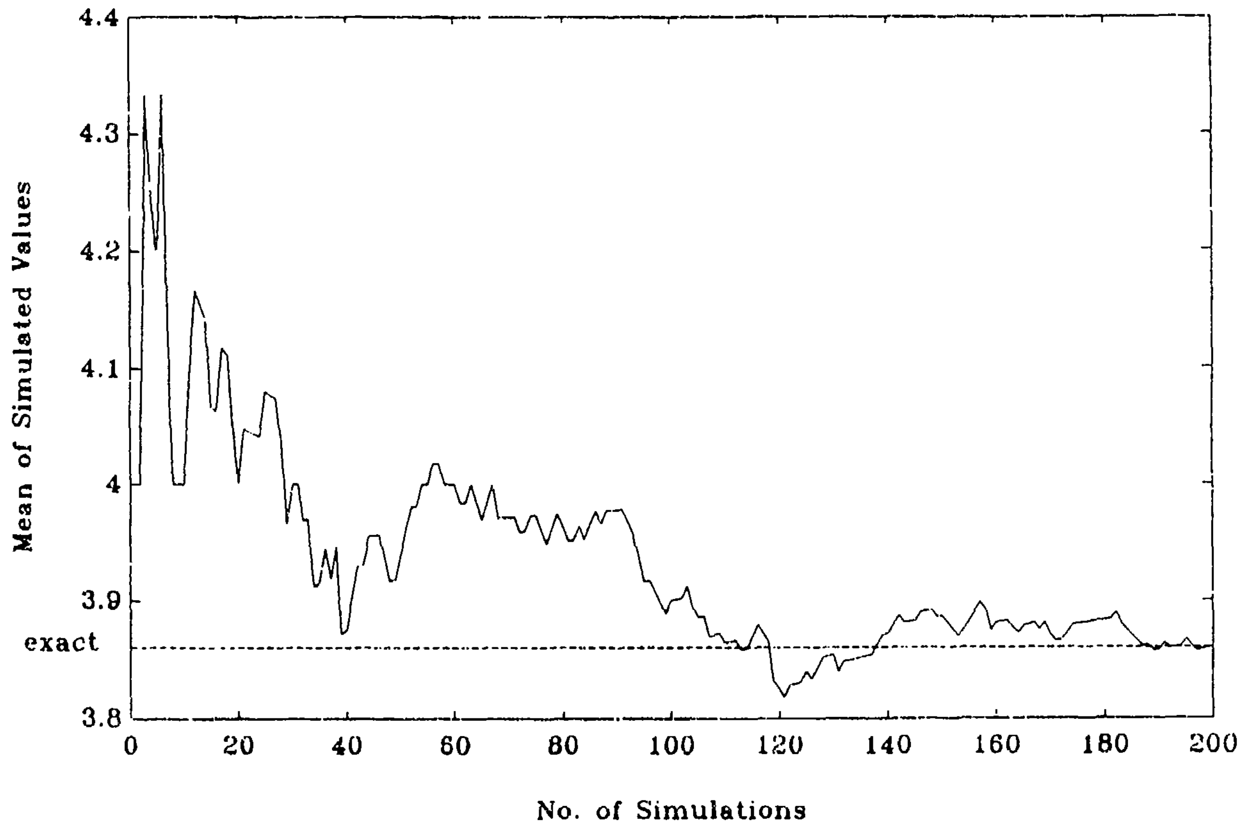


Figure 5.6: Mean of the values for the Monte Carlo simulations vs. the number of simulations performed for the baseline problem; all weapons are used in the first stage. The dashed line is the exact expected value for the problem.

of these Monte Carlo Simulations are performed (we have used 100 runs) and the sample mean is then taken as an approximation to the expected value of the assignment $\vec{X}$. We have found that after about 100 simulations the first two digits of the sample mean remain constant.

Consider, for example, the baseline problem. If few weapons are used in stage 1 then the number of possible outcomes is small and one would expect that after a few simulations the expected value would be obtained. Therefore as the number of weapons in the first stage increases (and the number in stage 2 decreases) the number of simulations necessary will increase. Let us consider the case in which all of the weapons are used in stage 1. This case should require the most number of simulations. Furthermore we can compute the expected value for this case exactly since it is simply a simulation of a static problem for which the expected value can be computed. In figure 5.6 we have plotted the sample mean of the values of the simulations versus the number of simulations. Note that in this case we do not need to use a lower bound for the optimal value in stage 2 since we can compute the optimal stage 2 value exactly. On the plot we have also included the exact expected value for the problem which is given as follows:

$$J_1 = 5(1 - (.4)^4)^10 = 3.86$$

Note that only five of the assets are defended; each are defended with 40 weapons. Note that after 100 simulations the first two digits of the sample mean is correct and remains correct.

The value for any given first stage assignment is a random variable. An important quantity is the variance of this random variable. This will indicate how close the value of any single outcome is likely to be to the expected value. In figure 5.7 we have plotted the sample standard deviation (the root of the variance) versus the number of simulations for the baseline problem in which all weapons are used in stage 1. Again since this is a static problem the exact value of the standard deviation can be computed as:

$$\sigma = [5(1 - (.4)^4)^10(1 - (1 - (.4)^4)^10)]^{\frac{1}{2}} = 0.94.$$

Note that after about 50 simulations the sample standard deviation is close to the exact value. Note that for this problem the expected value is 3.86 and the standard deviation is 0.94. The standard deviation is therefore roughly 25% of the optimal value. This means that any two outcomes can have significantly different values.

Let us now consider the baseline problem but use 100 weapons in each of the stages. In this case we cannot compute the expected value and the standard deviation exactly because of two reasons, (a) we do not have an optimal algorithm to compute the optimal value for the stage 2 problem and (b) the number of possible outcomes of stage 1 is too large. Because of these problems we will compute bounds on the expected value by simulating

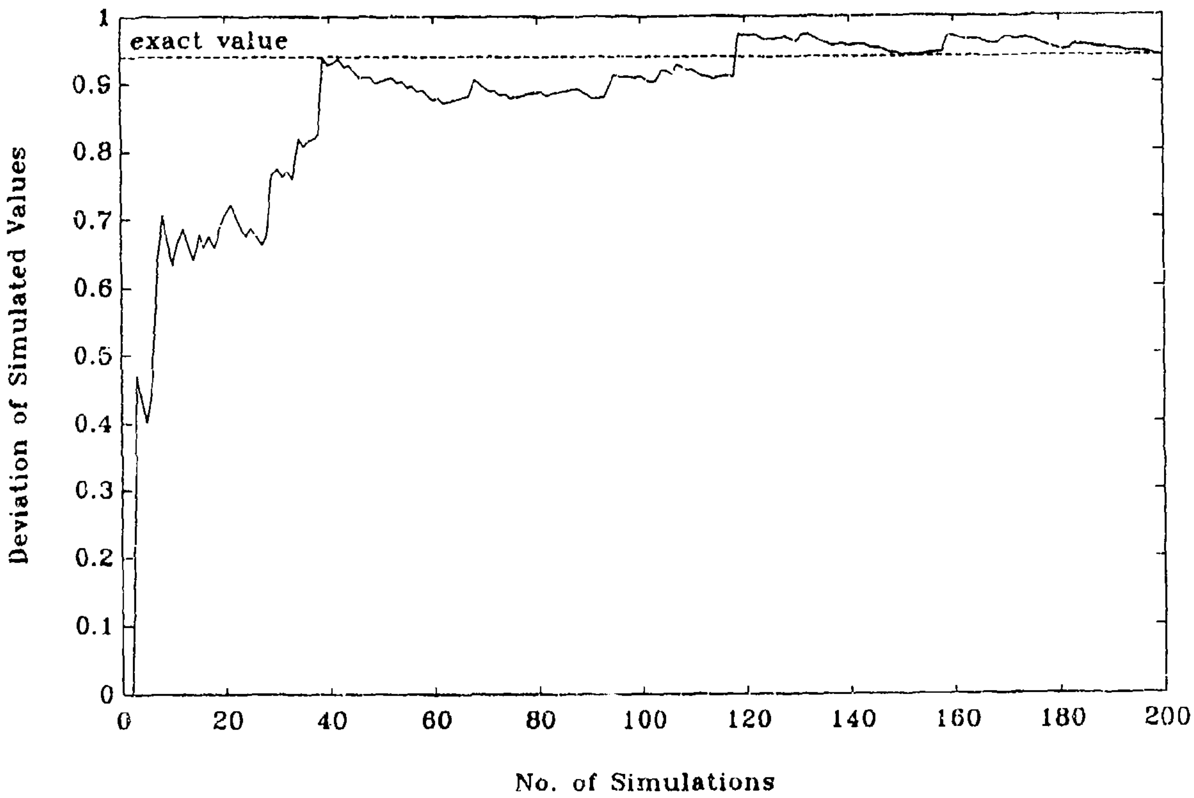


Figure 5.7: Standard Deviation of values for the Monte Carlo simulations vs. the number of simulations for the baseline problem; all weapons are used in the first stage. The dashed line is the exact standard deviation for the problem.

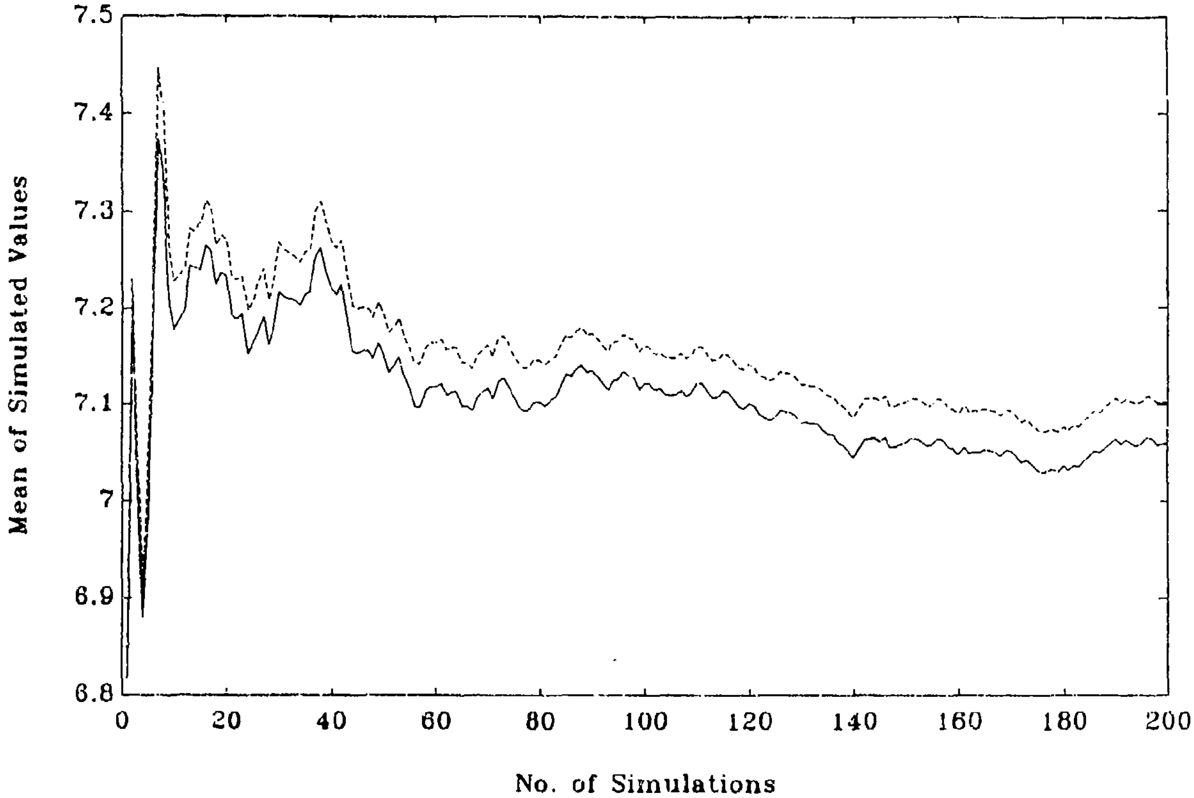


Figure 5.8: Mean of the values for the Monte Carlo simulations vs. the number of simulations performed for the baseline problem; 100 weapons are used in the each stage. The dashed line corresponds to using an upper bound for the second stage value. The solid line corresponds to using a lower bound for the second stage value.

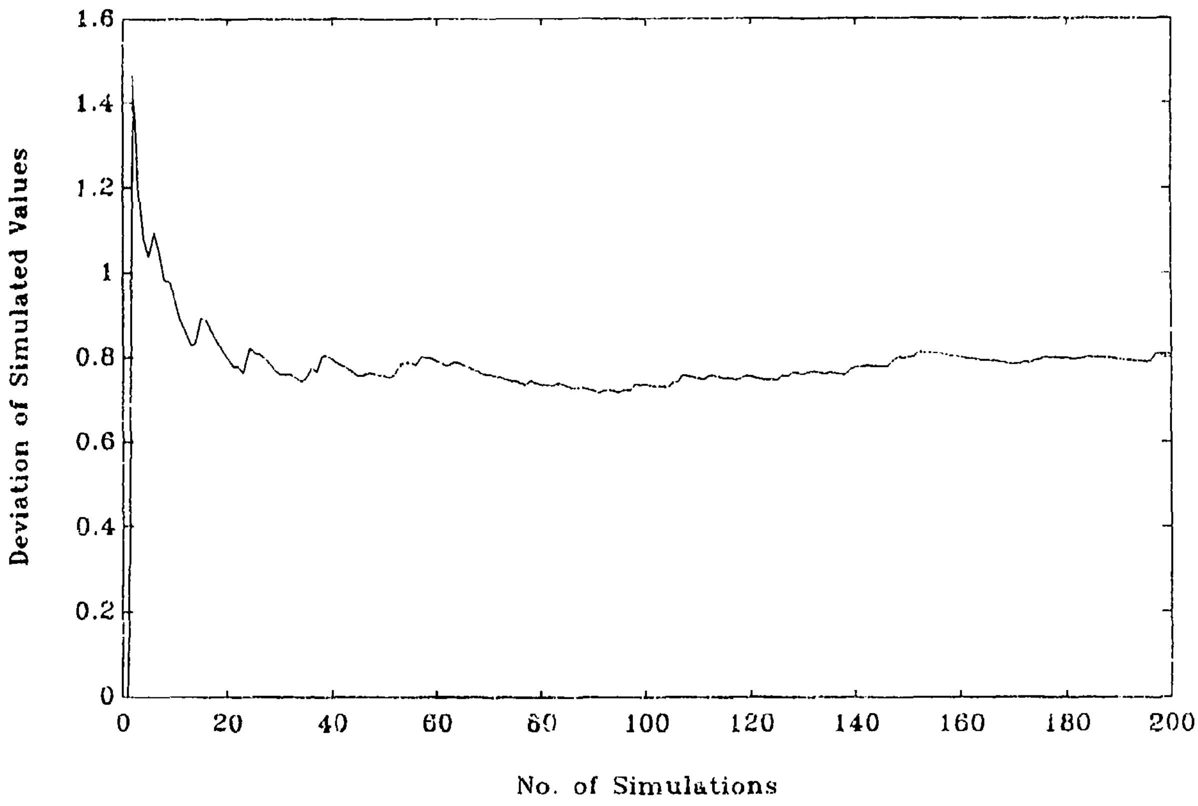


Figure 5.9: Standard Deviation of values for the Monte Carlo simulations vs. the number of simulations for the baseline problem; 100 weapons are used in each stage.

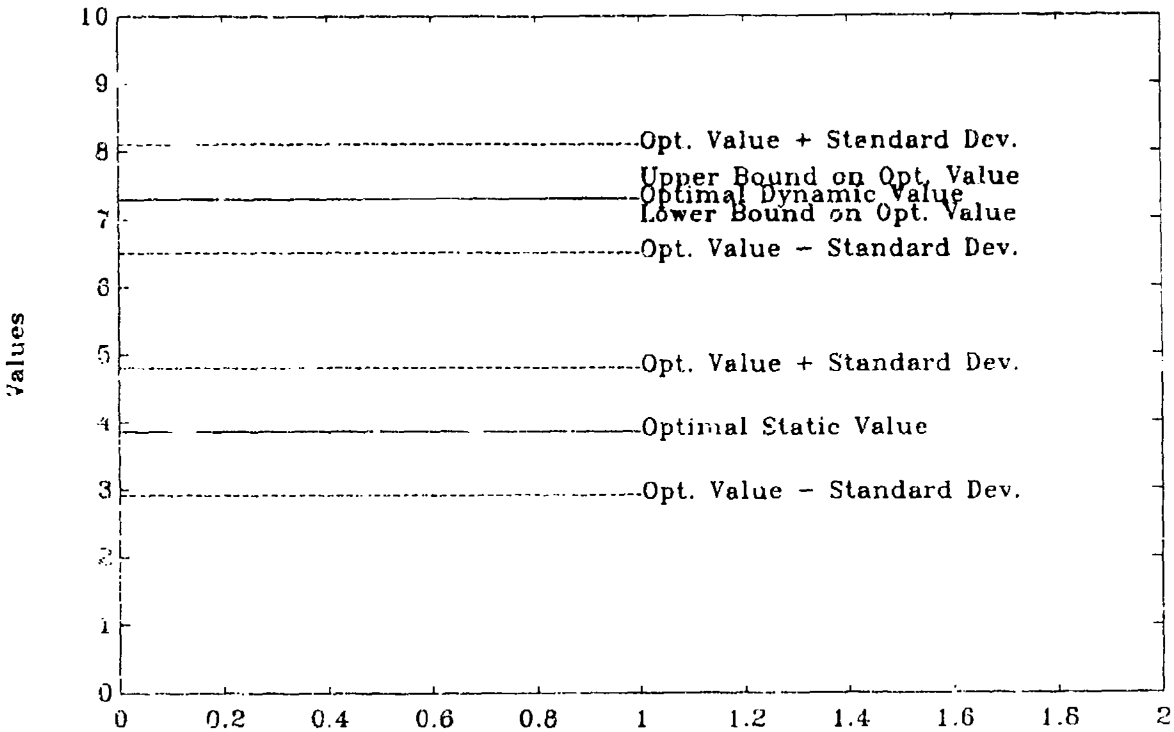


Figure 5.10: Various values for the dynamic and static strategies for the baseline problem

stage 1 and finding bounds for the expected value in stage 2. The standard deviation will be approximated by simulating stage 1 and then taking the sample deviation of the result. The bounds for the stage 2 optimal values are found by the use of the algorithm that was presented in chapter for the static strategy.

In figure 5.8 we have plotted the sample means of the upper and lower bounds. Note that each bound converges to some quantity and the difference of these quantities is small. In figure 5.9 we have plotted the sample standard deviation obtained by simulating stage 1.

Several values and bounds have been mentioned. In figure 5.10 we have shown where each of these quantities lies relative to the others. These quantities are for the baseline problem using 90 weapons in stage one which is the best ... obtained by our algorithm.

The upper bound on the optimal value for the dynamic strategy is 7.46. The lower bound on the value of the suboptimal solution of our algorithm is 7.12. Note that this provides us with a lower bound on the optimal value. For illustrative purposes let us assume that the optimal value of the dynamic strategy is 7.2. We will assume that the standard deviation for the optimal solution is the same as the standard deviation of the sub-optimal solution which we will take to be the sample deviation obtained after 200 simulations (see figure 5.9). We have also included the optimal static value which is 3.86. In general the optimal value for the static problem may not be computable but upper and lower bounds can be obtained. The standard deviation for the static value is also shown.

### 5.6.2 Discussion of Upper Bound Computation

If we fix the number of weapons to be used in stage 1 then one can obtain an upper bound on the optimal value (for that number of first stage weapons) of the problem by solving the dynamic problem with the upper bound approximation $\bar{J}$. In order to obtain the global solution of the problem one must search over all possible values of $m_1$, the number of weapons used in stage 1. In figure 5.11 we have plotted the lower bound on the value of the solution produced by the algorithm (solid line) as well as an upper bound on the optimal value (dashed line) versus the number of weapons used in stage 1. In order to obtain a lower bound on the optimal value of the problem we must choose the maximum over all values of $m_1$ of the solid line. To obtain an upper bound on the optimal value we must choose the maximum over all values of $m_1$ of the dashed line. Unfortunately we find that each of these functions peaks at different points. It is, however, very unlikely that the optimal value of $m_1$ is obtained at the peak of the upper bound because at that point the lower bound is extremely small. *We will therefore assume that the optimal value of $m_1$ is the point at which the lower bound on the expected value of the solution of the algorithm peaks.* This is a very reasonable assumption since we believe that the shape of this function represents very closely the shape of the optimal one.

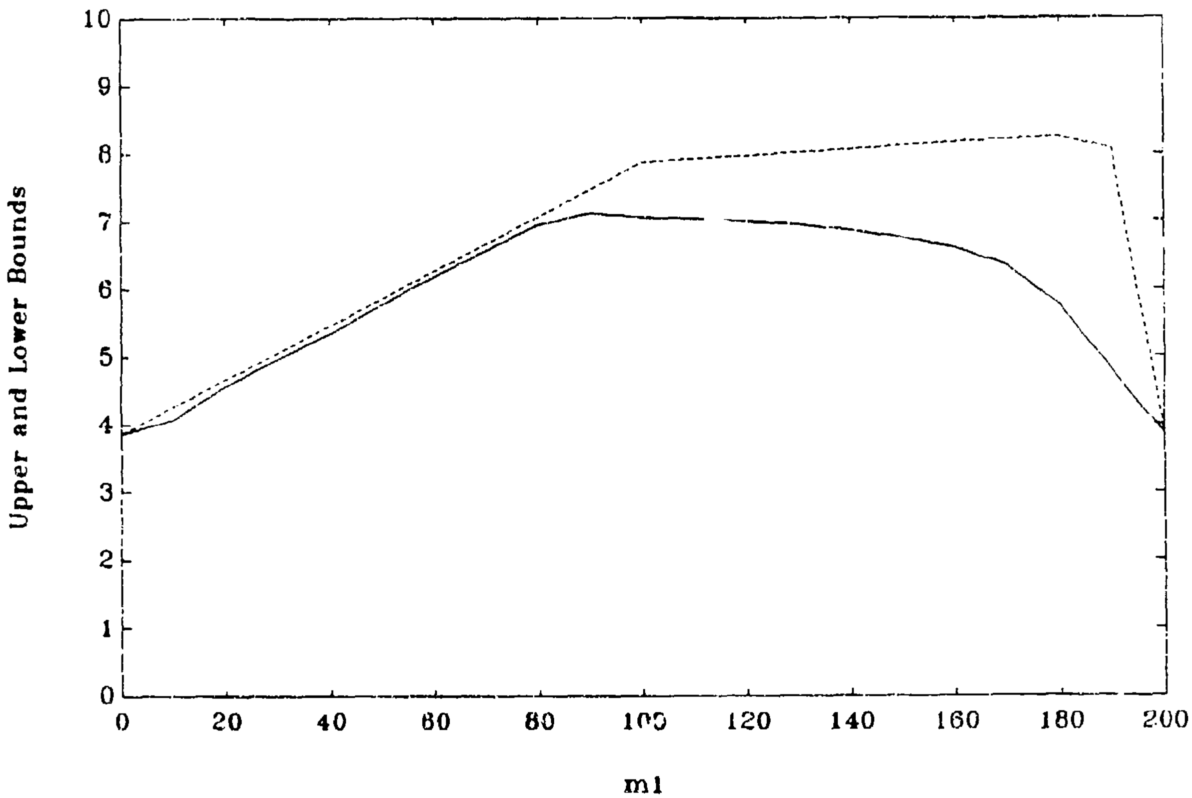


Figure 5.11: Upper and lower bounds on the optimal value vs. the number of weapons used in stage 1 for the baseline problem.

### 5.6.3 Sensitivity Analysis

One important feature that we have discovered about a dynamic strategy is that it is less sensitive to the number of assets defended compared to the static strategy. This can be illustrated as follows. In figure 5.12 we have plotted the lower bound on the expected value of the dynamic strategy versus the number of assets (uniformly) defended. We have also plotted the case for a static strategy. Note that for the dynamic strategy it makes little difference whether 8, 9 or 10 of the assets are defended in stage 1. On the other hand, the performance of the static strategy degrades significantly if the optimal number of assets are not defended. This difference will be of more importance in the general problem where one must decide on which subset of assets should be defended.

We will next present a simple but approximate expression for the optimal value of the two-stage dynamic strategy. We will assume that $n_k = n, W_k = W, \pi_k = \pi$ and $p_k(t) = p$. We have found empirically that a good approximation to the optimal value of $m_1$ under these assumptions is $M/2$. Let us suppose that it is optimal to defend $L$ of the assets in stage 1. The optimal strategy is to spread the weapons as evenly as possible among the $L$ assets. If this is done the expected number of targets aimed for an asset that survive stage 1 is given approximately by

$$\bar{n} \approx n(1-p)^{\frac{M}{2Ln}}.$$

Again we have found empirically that the assets defended in stage 2 are the same that were defended in stage 1. Also the weapons will again be spread as evenly as possible among the defended assets. If we assume that the expected number of targets survive stage 1 and that in stage 2 only the assets that were defended in stage 1 are defended then the cost can be approximated by

$$J_1^*(L) \approx L[1-(1-p)^{\frac{M}{2L\bar{n}}}]^{\bar{n}}. \tag{5.7}$$

We could not find a simple expression for the number of assets to be defended in stage 1. However, the expression 5.7 can easily be evaluated for different values of $L$ and the maximum value taken. Consider for example the baseline problem. If 8 assets are defended

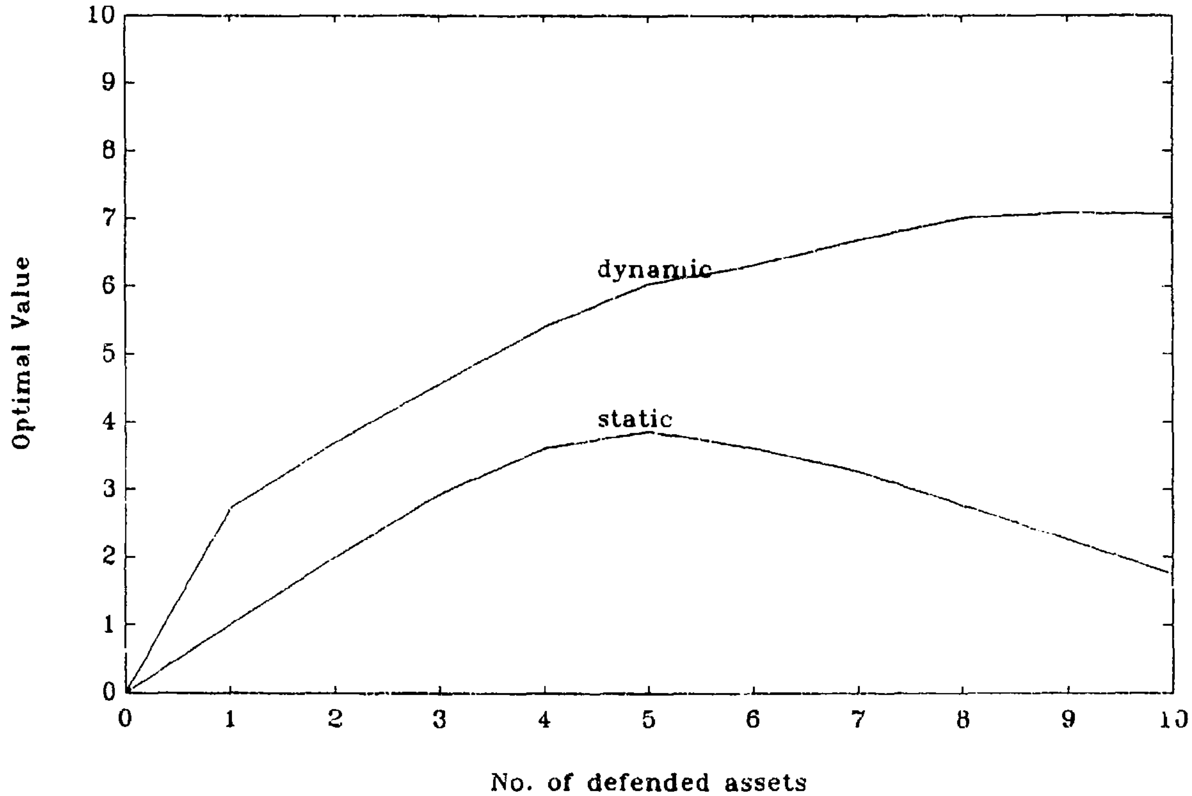


Figure 5.12: Optimal dynamic and static values as a function of the number of assets defended for the baseline problem

in stage 1 we get $\bar{n} = 3.18$ and so

$$J_1^*(8) \approx 8[1 - (.4)^{3.9}]^{3.2} = 7.3.$$

Similarly we can compute

$$J_1^*(9) \approx 7.2,$$

and

$$J_2^*(7) \approx 6.8.$$

Therefore it is better to defend eight of the assets than to defend seven or nine. The lower bound on the optimal value for this problem is 7.1 while the upper bound is 7.5. We therefore find that the approximation 7.3 is a good one. Using our heuristic we find that 9 of the 10 assets should be defended in stage one. The approximation says that 8 of the 10 assets should be defended in stage one.

Let us now consider the case of the baseline problem but with 100 defense weapons. If we use the above approximation we find that it is best to defend 4 of the assets in stage 1 and that the resulting value is given by

$$J_1^*(4) \approx 3.7.$$

The upper and lower bounds on the optimal value for this case is 3.93 and 3.47. So we again find that the solution obtained using the approxiamtion is very good.

These approximations can be used to get a rough estimate of the number of weapons to use in stage 1, the number of assets to defend in stage 1 and also the expected value of the assignment.

### 5.6.4 Numerical Examples

We will next consider what happens to the optimal solution and value as each of the parameters in the baseline problem is varied. Because of the uniformity of the problem, all but one of the assets will be defended by the same number of weapons. Therefore we will represent the solution of the dynamic problem (i.e. the stage 1 assignment) by the

| Kill probability | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|
| No. Stage 1 weapons | 150 | 90 | 100 | 100 | 100 | 100 |
| Dynamic Solution | 7.5 | 9 | 10 | 10 | 10 | 10 |
| Dynamic Upper Bound | 6.9 | 7.5 | 9.5 | 10 | 10 | 10 |
| Dynamic Lower Bound | 5.1 | 7.1 | 9.3 | 10 | 10 | 10 |
| Static Solution | 4 | 5 | 6.7 | 10 | 10 | 10 |
| Static Upper Bound | 2.9 | 3.9 | 5.1 | 6.6 | 9.0 | 10 |
| Static Lower Bound | 2.9 | 3.9 | 4.9 | 6.6 | 9.0 | 10 |
| Ratio Upper Bound | 2.4 | 1.9 | 1.9 | 1.5 | 1.1 | 1.0 |
| Ratio Lower Bound | 1.8 | 1.8 | 1.8 | 1.5 | 1.1 | 1.0 |

Table 5.1: Dynamic and Static Values for the Baseline problem with 200 weapons and various kill probabilities

| Kill probability | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|
| No. Stage 1 weapons | 100 | 80 | 90 | 100 | 100 | 100 |
| Dynamic Solution | 5 | 8 | 9 | 10 | 10 | 10 |
| Dynamic Upper Bound | 5.6 | 6.1 | 7.8 | 9.6 | 10 | 10 |
| Dynamic Lower Bound | 4.1 | 5.2 | 7.3 | 9.5 | 10 | 10 |
| Static Solution | 3 | 3.7 | 5 | 7.5 | 7.5 | 10 |
| Static Upper Bound | 2.2 | 2.9 | 3.8 | 5.0 | 6.8 | 10 |
| Static Lower Bound | 2.2 | 2.8 | 3.8 | 4.8 | 6.7 | 10 |
| Ratio Upper Bound | 2.6 | 2.1 | 2.0 | 2.0 | 1.5 | 1.0 |
| Ratio Lower Bound | 1.9 | 1.8 | 1.9 | 1.9 | 1.5 | 1.0 |

Table 5.2: Dynamic and Static Values for the Baseline problem with 150 weapons and various kill probabilities

| Kill probability | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|
| No. Stage 1 weapons | 50 | 50 | 60 | 70 | 90 | 100 |
| Dynamic Solution | 2.5 | 5 | 6 | 7 | 9 | 10 |
| Dynamic Upper Bound | 3.2 | 3.9 | 5.2 | 6.5 | 8.5 | 10 |
| Dynamic Lower Bound | 2.6 | 3.5 | 4.8 | 6.2 | 7.9 | 10 |
| Static Solution | 2 | 2.5 | 3.3 | 5 | 5 | 10 |
| Static Upper Bound | 1.5 | 1.9 | 2.5 | 3.3 | 4.5 | 10 |
| Static Lower Bound | 1.5 | 1.7 | 2.3 | 3.3 | 4.5 | 10 |
| Ratio Upper Bound | 2.1 | 2.3 | 2.3 | 2.0 | 1.9 | 1.0 |
| Ratio Lower Bound | 1.7 | 1.8 | 1.9 | 1.9 | 1.8 | 1.0 |

Table 5.3: Dynamic and Static Values for the Baseline problem with 100 weapons and various kill probabilities

number of assets defended in stage 1. All but one of the assets will be defended with the same number of weapons. The other defended asset will be defended with a lesser number of weapons so it will not be as heavily defended as the others. We will include this asset as a fraction in the number of assets defended. This fraction will be the ratio of the number of weapons assigned to defend the asset and the number of weapons assigned to each of the other (more heavily defended) assets. We also computed upper and lower bounds on the ratio of the optimal dynamic and static values. These bounds were obtained using the upper and lower bounds on the optimal Dynamic and Static values. In 5.1 we have included dynamic and static results for the baseline problem. Table 5.2 contains the dynamic and static results for the baseline problem with 150 weapons while table 5.3 contains results for the baseline problem with 100 weapons. We will next provide some examples and discuss the implications of each of the results.

In figure 5.13 we have plotted the upper and lower bounds on the optimal values for both the dynamic and static strategies for the baseline problem with 200 weapons (solid lines). We have also plotted the approximation obtained using equation 5.7 (dashed line). Note that the approximation is a good one. In figure 5.14 we have plotted the bounds for the baseline problem with 100 weapons. Note that the plots for the static and dynamic strategies have roughly the same shape (convex for small values of $p$ and concave for larger values). The only difference is that the curve for the static strategy is lower. Therefore let us consider the following. If we keep the number of weapons for the dynamic problem fixed and increase the number of weapons for the static problem, then the curve for the static problem will approach that for the dynamic problem. Some simple calculations show that this will occur when the number of weapons for the static problem is roughly twice that for the dynamic problem. We, therefore, find once more that the dynamic strategy requires about half as many weapons as the static strategy for the same level of performance. Recall that this result was also true for the Dynamic Target-Based problem. Also note that the optimal values for both problems are sensitive to the kill probability.

In figure 5.15 we have plotted the upper and lower bounds on the ratio of the optimal

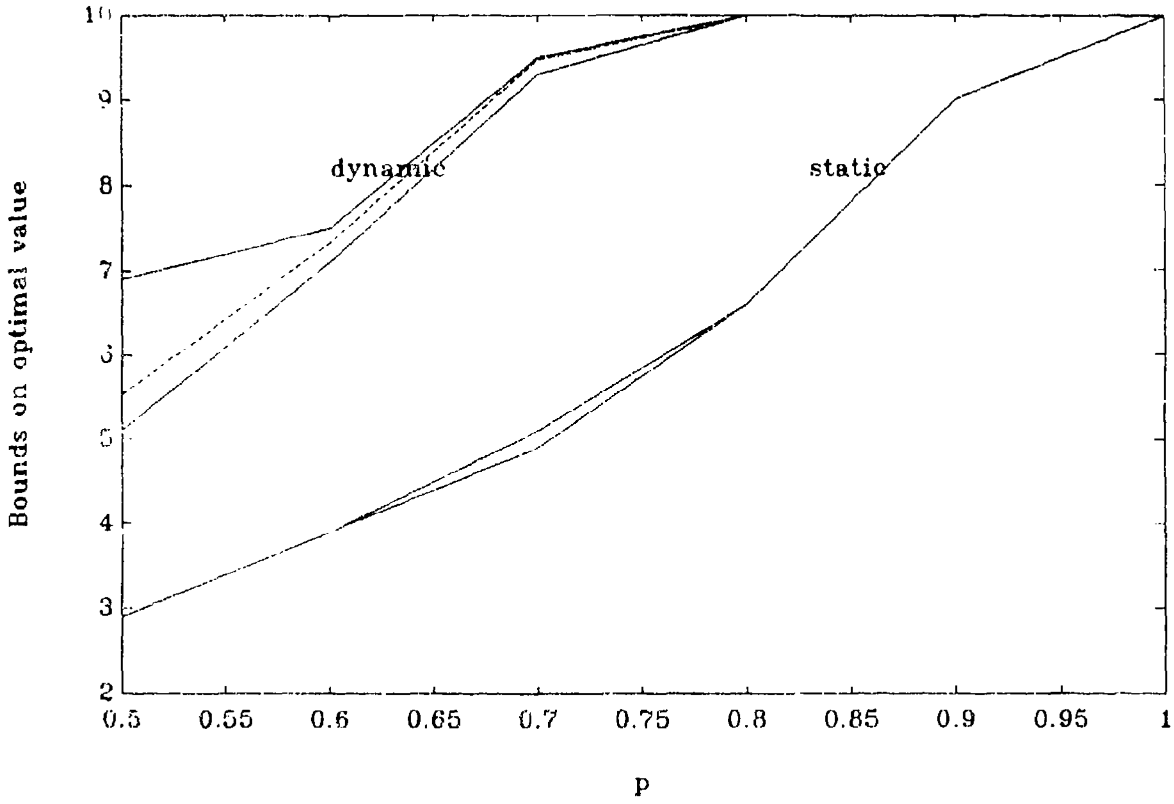


Figure 5.13: Upper and lower bounds on the optimal values for the dynamic and static problems versus the kill probability for the baseline problem with 200 weapons. The dashed line is the approximation to the optimal dynamic value obtained using equation 5.7.

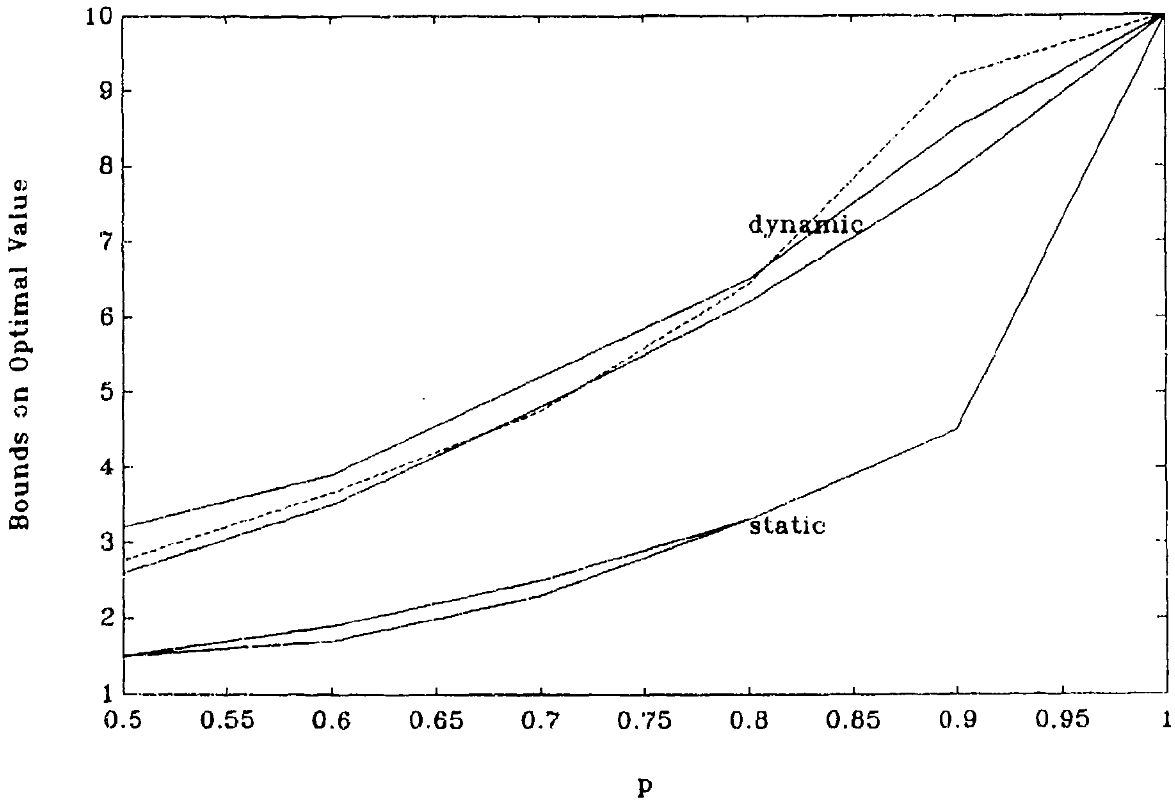


Figure 5.14: Upper and lower bounds on the optimal values for the dynamic and static problems versus the kill probability for the baseline problem with 100 weapons. The dashed line is the approximation to the optimal dynamic value obtained using equation 5.7.

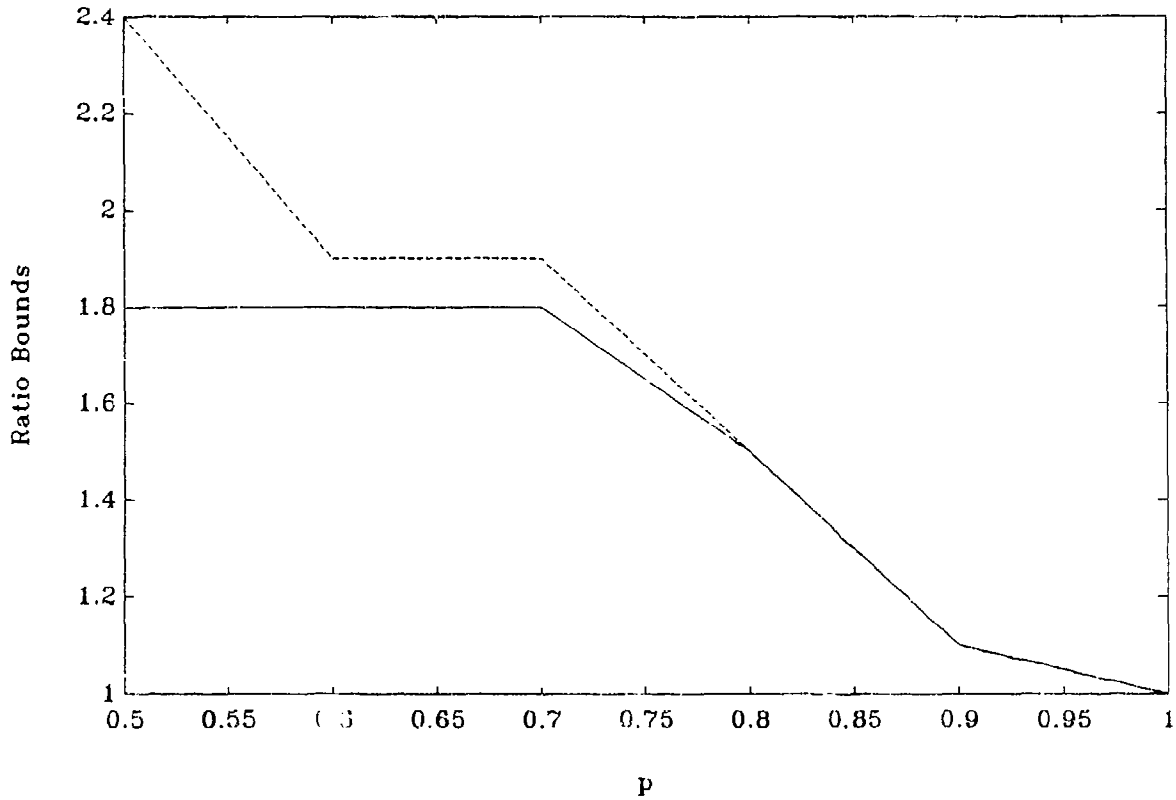


Figure 5.15: Upper and lower bounds on the ratio of the optimal values for the dynamic and static problems versus the kill probability for the baseline problem with 200 weapons.

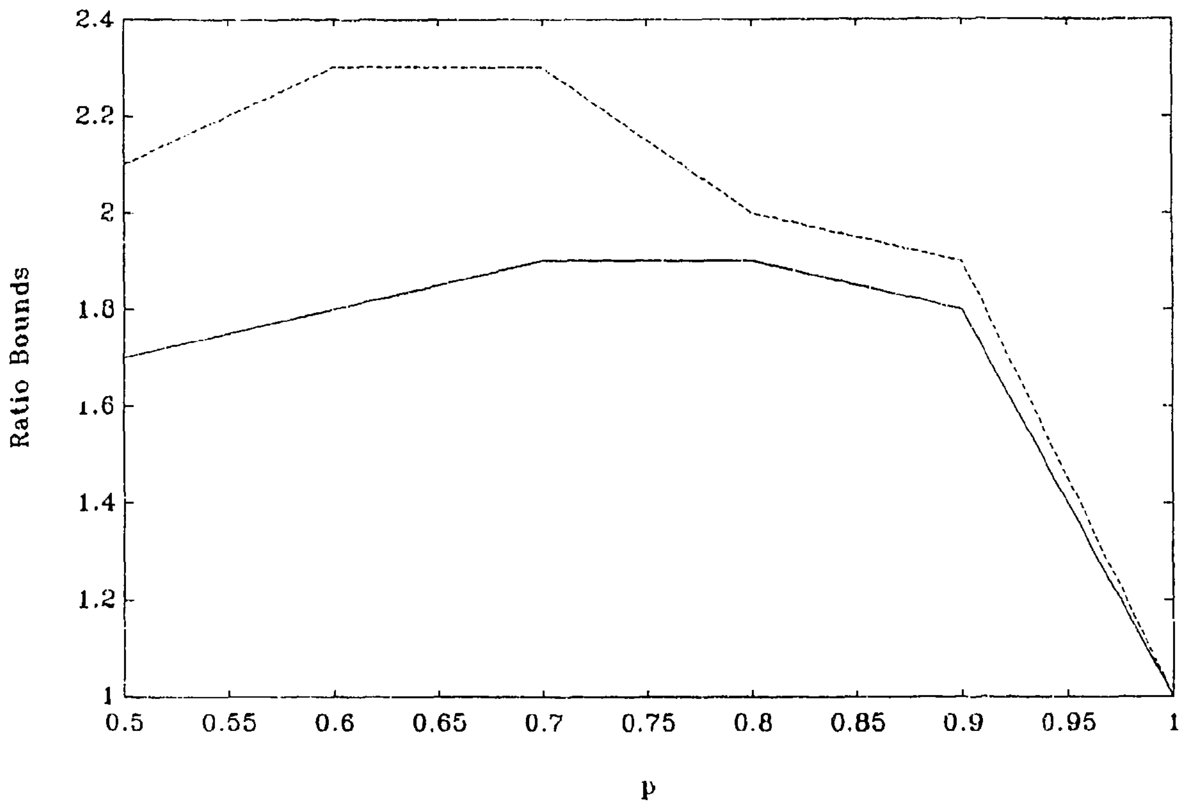


Figure 5.16: Upper and lower bounds on the ratio of the optimal values for the dynamic and static problems versus the kill probability for the baseline problem with 100 weapons.

dynamic and static values versus the kill probability for the baseline problem with 200 weapons. We have also included the value obtained by the approximation presented in the previous subsection (the dashed line). In figure 5.16 these bounds were plotted for the baseline problem with 100 weapons. Note that the ratio goes to unity as the kill probability goes to one. This happens because, in the limit of unit kill probabilities, all assets will be saved if either strategy is used. For kill probabilities less than 0.9 the ratio of the optimal dynamic and static strategies is roughly two. In other words, a dynamic strategy outperforms a static one by roughly a factor of two. Note that, for static problems, the optimal value increases linearly with the number of weapons. Therefore, twice as many weapons will be required for the static problem to obtain the same performance as the dynamic problem. This is the same conclusion that was drawn in the previous paragraph.

The only difference between the curve for the case of 200 weapons and that for the case of 100 weapons is that the former curve starts decreasing around $p = .8$ while the latter curve starts decreasing around $p = .95$. Since there are 100 weapons, there will be double shooting if 200 weapons are used. Note that the effect of double shooting with a kill probability of $p$ is the same as single shooting with a kill probability of $1 - (1-p)^2$). We will therefore consider the case of 200 weapons as being equivalent to the case of 100 weapons with a kill probability of $p(2-p)$. The following can therefore be used as a rule of thumb. If the number of "perfect" weapons ($M(1-(1-p)^{\frac{M}{N}})$ is less than 95 and greater than 50 then the ratio of the optimal values of the dynamic and static problems is roughly two. This means that, for these types of problems, the dynamic strategy outperforms the static one by a factor of two.

In the following tables we will investigate the affect of changing various parameters of the baseline problem. We will provide the results for both the dynamic and static problems for comparisons. Note that, for the baseline problem there are 10 unit valued assets. Therefore the expected surviving value of the assets cannot be greater than 10.

**Table 1: Baseline Problem**

The baseline problem: $M = 200, N = 100, T = 2, K = 10, n_k = 10, p_k = 0.6, \pi_k = 1, W_k = 1$ for $k = 1, \ldots, K$.

STATIC CASE:

Optimal solution[3]: [0,0,0,0,0,40,40,40,40,40]

Optimal value: 3.86

Standard Deviation: 0.94

DYNAMIC CASE:

Number of weapons used in first stage: 90

Assignment of these weapons[4] : [0,10,10,10,10,10,10,10,10,10]

Lower bound on value of this solution: 7.12

Upper bound on optimal value: 7.46

REMARKS:

Note that for the static case the optimal strategy is to defend half of the assets uniformly (preferential defense). For the dynamic case 9 of the assets are defended in the first stage. Note also that the value of the solution produced by the sub-optimal algorithm is close to the upper bound on the optimal value. This implies that the sub-optimal solution is either equal or close to the optimal solution.

Note that a typical stage 2 state might be [0,4,4,4,4,4,4,4,4,4]. For this state the optimal stage 2 solution would be [2,12,12,12,12,12,12,12,12,12]. Therefore we see that the same number of assets that were defended in stage 1 are defended in stage 2. This makes sense since weapons would be wasted if more assets were defended in stage 1 than in stage 2.

Finally note that the optimal dynamic value is roughly twice that of the optimal static value. This means that the defense can save roughly twice as many assets by using a dynamic strategy. Since roughly seven assets are eventually saved with the dynamic strategy then

[3]Represented by the number of weapons assigned to defend each of the 10 assets. The number of weapons assigned to each of the targets directed to an asset can be obtained by dividing by 10 the number of weapons assigned to the defense of that asset.

[4]Represented by the number of weapons assigned to defend each of the 10 assets in the first stage.

why does the defense attempt to save 9 assets in the first stage? Let us consider such a strategy. Consider the assignment in which 160 weapons are used in stage 1. These weapons are used to defend 8 assets with 20 weapons each. The expected value for this solution is 7.09. Therefore we find that if the defense did try to save 8 assets in the first stage then the resulting solution is near-optimal. This suggests that any reasonable strategy will be near-optimal.

**Table 2: Baseline Problem with lower kill probability**

The baseline probl m except that the kill probability for each weapon-target pair in each stage is 0.5:

STATIC CASE:

Optimal solution: [0,0,0,0,0,0,50,50,50,50]

Optimal value: 2.91

DYNAMIC CASE:

Number of weapons used in first stage: 150

Assignment of these weapons: [0,0,10,20,20,20,20,20,20,20]

Lower bound on value of this solution: 5.10

Upper bound on optimal value: 6.90

REMARKS:

In this case we note that even for the dynamic problem it is better to use a preferential defense in stage 1. However, 7.5 assets are defended in the dynamic case compared to 4 in the static case. Note that the algorithm is able to handle such cases for the dynamic problem.

**Table 3: Baseline Problem with higher kill probability**

The baseline problem except that the kill probability for each weapon-target pair in each stage is 0.7:

STATIC CASE:

Sub-optimal solution: [0,0,0,20,30,30,30,30,30,30]

Value of suboptimal solution: 4.95

Upper bound on optimal value: 5.07

DYNAMIC CASE:

Number of weapons used in first stage: 100

Assignment of these weapons: [10,10,10,10,10,10,10,10,10,10]

Lower bound on value of this solution: 9.35

Upper bound on optimal value: 9.47

REMARKS:

As the kill probability of the weapons increases we find that for the dynamic case all of the assets are defended. Note also that either the solution produced by the algorithm is getting closer to optimal or the upper bound on the optimal value is improving (or both) as the kill probability increases.

**Table 4: Baseline Problem with increasing (with stage) kill probabilities**

The baseline problem except that the kill probability of the weapons in the first stage is 0.5 while their kill probability in the second stage is 0.7:

STATIC CASE: (all weapons fired in stage 2)

Sub-optimal solution: [0,0,0,20,30,30,30,30,30,30]

Value of suboptimal solution: 4.95

Upper bound on optimal value: 5.07

DYNAMIC CASE:

Number of weapons used in first stage: 90

Assignment of these weapons: [0,10,10,10,10.10,10,10,10,10]

Lower bound on value of this solution: 6.89

Upper bound on optimal value: 7.22

REMARKS:

Our intuition for this case is that more weapons should be used in the stage with higher kill probability (stage 2) than in the other stage. The solution produced by the algorithm does in fact have this property. However, note that although the difference in the kill probabilities is large (0.5 and 0.7) only 20 more weapons are used in stage 2 than in stage 1.

**Table 5: Baseline Problem with decreasing (with stage) kill probabilities**

The baseline problem except that the kill probability of the weapons in the first stage is 0.7 while their kill probability in the second stage is 0.5. Note that this is the reverse of problem 4:

STATIC CASE: (all weapons fired in stage 1)

Sub-optimal solution: [0,0,0,20,30,30,30,30,30,30]

Value of suboptimal solution: 4.95

Upper bound on optimal value: 5.07

DYNAMIC CASE:

Number of weapons used in first stage: 120

Assignment of these weapons: [10,10,10,10,10,10,10,10,20,20]

Lower bound on value of this solution: 7.67

Upper bound on optimal value: 8.52

REMARKS:

Again note that we obtain the intuitive result that more weapons should be used in the stage with higher kill probability. However, if 100 weapons are used in stage 1 the lower bound on the value of the resulting solution is 7.62. Therefore the value does not seem to be very sensitive to the number of weapons used in stage 1. Finally note that the optimal value for this case is approximately 8.1 while that for the previous problem is approximately 7.1. Therefore we find that it is better to use the more effective weapons in stage 1 rather than in stage 2. A similar result was also obtained for the Dynamic Target-Based problem.

**Table 6: Baseline Problem with less weapons**

The baseline problem except that the defense has 100 weapons:

STATIC CASE:

Sub-optimal solution: [0,0,0,0,0.0,0,20,40,40]

Value of suboptimal solution: 1.72

Upper bound on optimal value: 1.93

DYNAMIC CASE:

Number of weapons used in first stage: 50

Assignment of these weapons: [0,0,0,0,0,10,10,10,10,10]

Lower bound on value of this solution: 3.47

Upper bound on optimal value: 3.93

REMARKS:

Here we find that the dynamic strategy performs better than the static one even if the number of weapons equals the number of targets. Again we find that the weapons should be divided equally between the stages. Also note that the performance of the dynamic strategy is approximately twice that of the static one as we have found for most of the problems. Also recall that for the Dynami Target-Based problem if the number of weapons was less than or equal to the number of targets then a dynamic strategy could not perform any better than a static one. This is a major difference between the two problems.

**Table 7: Baseline Problem with more weapons**

The baseline problem except that the defense has 300 weapons:

STATIC CASE:

Sub-optimal solution: [0,0,20,40,40,40,40,40,40,40]

Value of suboptimal solution: 5.58

Upper bound on optimal value: 5.79

DYNAMIC CASE:

Number of weapons used in first stage: 200

Assignment of these weapons: [20,20,20,20,20,20,20,20,20,20]

Lower bound on value of this solution: 9.88

Upper bound on optimal value: 10.00

REMARKS:

The bound on the optimal value for the dynamic value obtained using our algorithm was actually 10.29. However since there are only 10 assets, each of unit value, the maximum possible value is 10. We therefore find that the algorithm could produce a useless bound as in this case. However we have found that for the cases in which this occurs a good upper bound is the total sum of the asset values. Also note that in this case 200 weapons are used in the first stage. If 150 weapons are used in the first stage the lower bound on the resulting solution is 9.73. Therefore if half of the weapons are used in the first stage as was the case in most of the other problems the resulting value is still near-optimal.

**Table 8: Baseline Problem with less targets per asset**

The baseline problem except that there are 20 assets each of unit value with 5 targets aimed at each asset:

STATIC CASE:

Sub-optimal solution: [0,0,0,0,0,0,5,15,15,...,15]

Value of suboptimal solution: 9.42

Upper bound on optimal value: 9.58

DYNAMIC CASE:

Number of weapons used in first stage: 100

Assignment of these weapons: [5,5,...,5,5]

Lower bound on value of this solution: 16.35

Upper bound on optimal value: 16.61

REMARKS:

In this case 82% of the asset value is saved while for the baseline problem 70% was saved This indicates that smaller attacks on each asset favors the defense. This was also true for the static problem. In other words if the number of assets is kept fixed then as the number of targets increases, the performance of the defense decreases even if the weapon to target ratio was kept fixed. Therefore if the defense wishes to maintain the same performance it must increase its arsenal at a greater rate than that of the offense.

**Table 9: Baseline Problem with more assets**

The baseline problem except that there are 15 assets of unit value and the defense has 300 weapons:

STATIC CASE:

Sub-optimal solution: [0,0,0,0,0,0,0,20,40,40,40,40,40,40,40]

Value of suboptimal solution: 5.58

Upper bound on optimal value: 5.79

DYNAMIC CASE:

Number of weapons used in first stage: 170

Assignment of these weapons: [10,10,10,10,10,10,10,10,10,10,10,10,10,20,20]

Lower bound on value of this solution: 10.57

Upper bound on optimal value: 11.90

REMARKS:

In this case we have increased the number of assets while keeping the weapon to target ratio fixed. We find that the percentage of asset value saved in this case (70%) is approximately the same as that for the baseline problem (71%). We also find that the fraction of weapons used in the first stage is closer to half than for the baseline problem with 300 weapons. It therefore appears that as the size of the problem increases this fraction tends towards one half. Finally note that if 150 weapons are used in the first stage the lower bound on the value of the solution is 10.54. This again shows that using half of the weapons in the first stage results in a near-optimal solution.

**Table 10: Baseline Problem with higher kill probability but less weapons**

The baseline problem: except that the kill probability of each weapon-target pair in each of the stages is 0.8 and the defense has 100 weapons:

STATIC CASE:

Optimal solution: [0,0,0,0,0,20,20,20,20,20]

Optimal value: 3.32

DYNAMIC CASE:

Number of weapons used in first stage: 70

Assignment of these weapons: [0,0,0,10,10,10,10,10,10,10]

Lower bound on value of this solution: 6.25

Upper bound on optimal value: 6.53

REMARKS:

For this problem we have decreased the number of weapons but increased their kill probability. Note that although there are few weapons the dynamic strategy can still make more effective use of them than the static one. Also note that we can consider the defense as having 80 perfect weapons ($Mp$). If we look at the baseline problem with a kill probability of 0.5 then the defense can be considered as having 100 weapons. However the optimal value for the former problem is about 6.4 while that of the latter is about 5.7. This indicates that looking at the problem in these terms (i.e. perfect weapons) can be ery misleading. However, since 200 weapons are used for the baseline problem and there e 100 targets let us consider the equivalent kill probability if a target is double shot. Since $p = .5$ then the equivalent kill probability of two weapons is 0.75. This corresponds to 75 perfect weapons. Using this approach we find that the baseline problem with a kill probability of 0.5 should perform worse and indeed it does.

**Table 11: Baseline problem with different asset values**

The baseline problem except that the asset values are given by $\vec{W} = [1,1,1,1,1,3,3,3,3,3]$. Note that the maximum possible expected value is 20.

STATIC CASE:

Optimal solution: [0,0,0,0,0,40,40,40,40,40]

Optimal value: 11.57

DYNAMIC CASE:

Number of weapons used in first stage: 110

Assignment of these weapons: [0,0,0,0,10,20,20,20,20,20]

Lower bound on value of this solution: 16.19

Upper bound on optimal value: 17.94

REMARKS:

Note that the optimal solution of the static problem is the same as for the baseline problem. Since all of the larger valued assets are defended, the optimal value is three times that for the baseline problem. On the other hand the optimal solution for the dynamic case is to defend all of the larger valued assets with 20 weapons each and to defend one of the unit valued assets with 10 weapons. Recall that in the baseline problem 9 of the assets were defended in the first stage. Note that in this case 81% of the total asset value is saved compared to 71% for the baseline problem. This is expected because, since this problem is non-uniform, the lower valued assets can be left undefended when a preferential defense is used.

**Table 12: Baseline problem with different kill probabilities**

The baseline problem except that the kill probabilities in each stage is given by $\vec{p}(t) = [.5,.5,.5,.5,.5,.68,.68,.68,.68,.68]$.

STATIC CASE:

Optimal solution: [0,0,0,0,0,40,40,40,40,40]

Optimal value: 4.50

DYNAMIC CASE:

Number of weapons used in first stage: 90

Assignment of these weapons: [0,10,10,10,10,10,10,10,10,10]

Lower bound on value of this solution: 6.94

Upper bound on optimal value: 7.23

REMARKS:

Note that the kill probabilities were chosen so $(1-.5)(1-.68)=(1-.6)^2$. In other words double shooting in the baseline problem is equivalent in lethality to double shooting in this problem with one low kill probability weapon and one high kill probability weapon. Note that the performance of the static case is better than the performance of the static case for the baseline problem. On the other hand the performance for the dynamic case is roughly the same as that for the baseline problem. Therefore the effect of differing kill probabilities is smaller in the dynamic problem.

**Table 13: Baseline problem with different targets per asset**

The baseline problem except that the number of targets aimed at each asset is given by $\vec{n}=[5,5,5,5,5,15,15,15,15,15]$.

STATIC CASE:

Optimal solution: [5,5,5,5,5,0,0,0,0,75]

Optimal value: 5.61

DYNAMIC CASE:

Number of weapons used in first stage: 100

Assignment of these weapons: [5,5,5,5,5,0,0,15,30,30]

Lower bound on value of this solution: 7.65

Upper bound on optimal value: 7.81

REMARKS:

The performance for the dynamic case is better than that for the baseline problem. Again this is due to the fact that the number of targets per asset is not the same for all assets but the average number of targets per asset is the same as for the baseline problem. Therefore when a preferential defense is required the assets with many targets aimed for them will be left undefended while the others would be defended.

## 5.7 Concluding Remarks

In this chapter we have presented the Dynamic Asset-Based problem together with a sub-optimal algorithm for finding a good solution. We have also presented a method for obtaining an upper bound on the optimal value. In our numerical results we have presented examples which illustrate various properties of the solution of the dynamic problem. We also performed comparisons of the dynamic and static strategies. The main conclusions can be summarized as follows:

- This is the most general of the problems considered and hence the most difficult. It is also NP-Complete.

- Because of the difficulty of the problem it is necessary to use approximations. Furthermore, the value of an assignment cannot, in practice, be evaluated exactly because of the number of operations required. Therefore this value must be estimated with the use of simulations.

- The sub-optimal algorithm presented performed well on the problems on which it was run. We believe that if this method is used on the more general version of the problem it will also perform well.

- The Target-Based approximation approach to solving the problem is advantageous because it can make use of algorithms that have already been developed for the Dynamic Target-Based problem. However in its pure form (i.e. without any modifications), this approach may perform poorly on problems for which a strong preferential defense is optimal.

- In general we have found that the performance of a dynamic strategy is roughly twice that for the corresponding static strategy. An equivalent statement is that half as many weapons are required for the dynamic problem to obtain the same level of performance as the static one. These results show the importance of using a dynamic approach. The increase in the computational complexity can be reduced by using approximations. We have found that simple approximations reduce the computational complexity while only slightly degrading the performance.

There are several directions in which one may pursue the research of this chapter. One can investigate the effect of kill probabilities which are dependent on both the weapon as well as the target. There is also the problem of evaluation of an assignment. We used simulations to do this but there may be other approaches. Since the assets and the weapons will be geographically distributed one should also study distributed algorithms for the problem.

# Chapter 6

# Summary and Conclusions

## 6.1 Summary

In this thesis we have considered a class of dynamic, nonlinear weapon-target allocation problems. In particular we have studied the Static Target-Based, the Dynamic Target-Based, the Static Asset-Based and the Dynamic Asset-Based Weapon-Target Allocation Problems. The main application of these problems is in military defense models. Our intent was to provide an intuitive understanding of the problems and their solutions.

The Static Target-Based WTA Problem was presented in chapter 2. In general this problem is NP-Complete. However, in the case of a single class of weapons, the optimal solution can be found efficiently. Lower bounds on the optimal solution of the general problem can be found by relaxing the integrality constraints of the decision variables and solving the dual of this problem. This bound is very helpful in judging heuristics for solving the more general version (i.e. with many weapon classes) of the problem since it can be used to estimate how close the heuristic solution is to the optimal one.

The Dynamic Target-Based WTA problem was presented in chapter 3. We have found that this is a significantly more difficult problem than the static version. Under the assumption of a single class of weapons, two decisions must be made, the optimal number of weapons to be used in each stage and the optimal assignment of these weapons to targets. The former problem is difficult because many local optima may exist so that basically a global search has to be done. The latter assignment problem is also difficult because of the analytical and computational complexity. However, for the case of two targets we have shown that the Maximum Marginal Return algorithm produces the optimal solution. We have also looked at other special cases of this problem for which optimal solutions were obtainable. The costs of these solutions were then compared with the costs of the solutions of the static problem. It was found that the defense can essentially double its arsenal by using a dynamic strategy rather than a static one. Several analytical results were also obtained for simple cases of the problem. These results have provided us with valuable insight into the problem.

In chapter 4 we presented the Static Asset-Based WTA problem. This problem is more difficult than the static Target-Based problem primarily because the objective function is non-concave. Under the assumption of a single class of weapons, we proposed a sub-optimal algorithm for the problem. This algorithm also produces an upper bound on the optimal value and in many cases it produces the optimal solution for the problem. Experimental results have shown that, for most practical purposes, the solution produced by the algorithm is optimal. We also presented several sensitivity analysis results.

The dynamic version of the Asset-Based WTA problem was presented in chapter 5. This is the most general of all the problems considered and hence the most difficult. We presented a sub-optimal algorithm for solving this problem under some simplifying assumptions. A method for finding an upper bound on the optimal value was also given. Examples were then presented to illustrate various properties of the solution of the problem. It was found that the solution obtained by the sub-optimal algorithm was near-optimal for many of the examples considered. Comparisons of the optimal value of the dynamic problem were made with those of the static problem. These comparisons suggest that the performance of a dynamic strategy is roughly twice that of the static one.

Our work has provided us with valuable insights into the class of problems that was studied. From our results we were able to make conclusions which will help direct the future of research into these problems. These will next be given.

Several heuristics are available for the solution of the Static Target-Based problem. We believe that heuristics based on the Maximum Marginal Return algorithm are both efficient and provide good solutions to the problem. We have proposed an optimal local search algorithm for the problem under the assumption of weapon independent kill probabilities. This algorithm also has the nice property that it can be implemented on a parallel computer. This is an important property for the military defense applications since the problem has to be solved in real time.

We believe that a Maximum Marginal Return algorithm will also work well for the Dynamic Target-Based problem. We have shown that such an algorithm is optimal in the case of weapon independent kill probabilities with two targets. For some simple cases we have the following interesting result. If the number of stages is large then the same optimal cost as that for the static problem can be obtained by using half the number of weapons for the corresponding dynamic version of the problem. In other words, the defense can essentially double its arsenal by going to a dynamic strategy. Note that, in terms of cost performance this corresponds to a much greater increase in performance than a factor of two. These results indicate that their is a great advantage in using a dynamic strategy rather than a static one. The greater computational complexity of the dynamic strategy compared to the static one can be greatly reduced by using simple approximations.

We have proposed a sub-optimal algorithm for the Static Asset-Based WTA problem which is, for most practical purposes, optimal. This result is supported by numerical evidence. We believe that this algorithm can be used as a basis for a heuristic for the more general version of the problem. For the case of military defense problems we believe that the Asset-Based version should be given more attention since it more accurately models the later stages of an attack. It also captures the idea of a preferential defense which makes intuitive sense.

The Dynamic Asset-Based WTA problem has been the most general problem we have considered. Numerical experimentation has shown that our sub-optimal algorithm performs well. This algorithm can be used as a basis for the more general version of the problem. Numerical results show that the dynamic strategy performs significantly better than a static one. For the problems we have considered the dynamic strategy had a cost performance which was roughly twice that of the static one. The large computational complexity of the dynamic problem can be significantly reduced by the use of simple proximations. Such approximations were used in our algorithm. We believe that such approximations only slightly reduce the cost performance for the dynamic strategy. Since the cost performance can be doubled with only a small increase in the computational complexity of the problem we conclude that research should be concentrated on dynamic versions of the problem.

## 6.2 Directions for Further Research

There are several directions in which further research can be done. We will mention some of these in this section.

The Target-Based WTA problem is a good model for the problem of military defense for the early stages of the attack during which the defense does not know the intent of the enemy. On the other hand, the Asset-Based problem is more applicable for the later stages of the attack when the defense knows the detailed objectives of the offensive weapons. An interesting question is how these two models can be combined into a dynamic model in which the early stages are Target-Based while the later stages are Asset-Based. One must also decide how many stages etc. should be considered. Furthermore, in a realistic scenario the attack will not be simultaneous as we have assumed. How then can a sequential attack be appropriately included? And how should the stage variables be defined? These questions suggest that there are many interesting problems in deciding on an appropriate realistic dynamic model for the problem.

Another issue that must be considered is that of coordination of the defense's weapons. These weapons will be geographically distributed and will be controlled by different computers. Questions that should be considered are: How often should these computers exchange information, what information should they exchange, how much information should be exchanged, what type of network architecture should be used for communication, which sensor information should be sent to which computer etc.? There are several questions to be asked and answered. The answers to these questions will depend on the weapons allocation algorithm being used.

As we have seen from our simple examples the computational complexity of the Weapon-Target Allocation problem is enormous. Even the static version of the Target-Based problem is NP-Complete. Furthermore these assignments must be done in real time since the duration of the attack may be short. In order to obtain near-optimal solutions as fast as possible one would have to resort to parallel computers. In this case weapon-target allocation algorithms which are easily parallelizable will have to be chosen. Such algorithms should therefore be studied.

Another area of research which should be considered is the simulation of these algorithms. In a realistic scenario there are several physical constraints on the problem which should be taken into account. We believe that when this is done the number of feasible solutions will be reduced significantly. One must therefore ask which of these constraints should be considered to simplify the optimization problem. Several other effects should also be included into the model such as the effect of a weapon-missile engagement on other weapons and missiles.

One can see from the above that to properly model a realistic scenario one must include several additional factors. The aim of this thesis was to look at the basic underlying problem to gain some insight. We believe that a similar approach should be used to gain insight for the additional factors mentioned above. These results can then be combined to produce good heuristics.

# Appendix A

# Notation and Definitions

This appendix contains notation and definitions used throughout the thesis.

## A.1 Notation

The following standard notation will be used in the thesis:

| | | |
|---|---|---|
| $\lceil x \rceil$ | $\stackrel{\text{def}}{=}$ | The smallest integer greater than or equal to $x \in \Re$, |
| $\lfloor x \rfloor$ | $\stackrel{\text{def}}{=}$ | The largest integer smaller than or equal to $x \in \Re$, |
| $\aleph$ | $\stackrel{\text{def}}{=}$ | The set of natural numbers, |
| $\Re$ | $\stackrel{\text{def}}{=}$ | The set of real numbers, |
| $Z_+$ | $\stackrel{\text{def}}{=}$ | The set of non-negative integers, |
| $Z_+^n$ | $\stackrel{\text{def}}{=}$ | The set of ordered $n$-tuples of non-negative integers, |
| $\vec{x}$ | $\stackrel{\text{def}}{=}$ | $[x_1, x_2, \ldots, x_n]^T$, |
| $e_i$ | $\stackrel{\text{def}}{=}$ | The $i^{\text{th}}$ column of the identity matrix, |
| $E(x)$ | $\stackrel{\text{def}}{=}$ | The expected value of the random variable $x$, |
| $\lvert S \rvert$ | $\stackrel{\text{def}}{=}$ | The size of the finite set $S$, |
| $\lvert\lvert \vec{x} \rvert\rvert$ | $\stackrel{\text{def}}{=}$ | $[\sum_{i=1}^{n} \lvert x_i \rvert^2]^{\frac{1}{2}}$, |
| $\binom{n}{k}$ | $\stackrel{\text{def}}{=}$ | $\frac{n!}{(n-k)!k!}$. |

## A.2 Definitions

In this section we will present definitions of concepts which are used throughout the thesis. Some references on this material are [13], [16] and [20].

The following are some basic definitions from convex analysis.

**Definition A.1** *Given two points $x, y \in \Re^n$, a* **convex combination** *of them is any point of the form*

$$z = \lambda x + (1 - \lambda)y, \qquad \lambda \in \Re \quad \text{and} \quad 0 \leq \lambda \leq 1.$$

*If $\lambda \neq 0, 1$ we say $z$ is a* **strict convex combination** *of $x$ and $y$.*

**Definition A.2** *A set $S \subseteq \Re^n$ is* **convex** *if it contains all convex combinations of pairs of points $x, y \in S$.*

**Definition A.3** *Let $S \subseteq \Re^n$ be a convex set. The function $f : S \longrightarrow \Re$ is* **convex** *in $S$ if for any two points $x, y, \in S$*

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \qquad \lambda \in \Re \quad \text{and} \quad 0 \leq \lambda \leq 1.$$

**Definition A.4** *A function $f$ defined in a convex set $S \subseteq \Re^n$ is called* **concave** *if $-f$ is convex in $S$.*

**Definition A.5** *A function $F : \Re \longrightarrow \Re$ to be minimized is said to be* **unimodal** *over the interval $[a, b]$ if there exists a $\bar{\lambda}$ that minimizes $F$ over the interval and for $\lambda_1, \lambda_2 \in [a, b]$ such that $F(\lambda_1) \neq F(\bar{\lambda})$, $F(\lambda_2) \neq F(\bar{\lambda})$, and $\lambda_1 < \lambda_2$ we have*

$$\lambda_2 \leq \bar{\lambda} \qquad \text{implies} \quad F(\lambda_1) > F(\lambda_2),$$

$$\lambda_1 \geq \bar{\lambda} \qquad \text{implies} \quad F(\lambda_1) < F(\lambda_2).$$

We will next introduce some of the terminology used in optimization theory. The problems to be considered in this thesis have the following general form:

$$\text{minimize} \quad f(x)$$

$$\text{subject to} \quad x \in S$$

The function $f : \Re^n \longrightarrow \Re$ is called the *objective function*. The set $S \subseteq \Re^n$ is called the *constraint set* and the elements of the set $S$ are called *feasible solutions.*

**Definition A.6** *The point $x^* \in S$ is called a* **local minimum** *of $f$ over $S$ if for some $\varepsilon > 0$*

$$f(x^*) \leq f(x) \qquad \forall x \in S \quad \text{such that} \quad ||x - x^*|| < \varepsilon.$$

**Definition A.7** *The point $x^* \in S$ is called a* **global minimum** *of $f$ over $S$ if*

$$f(x^*) \leq f(x) \qquad \forall x \in S.$$

We next give some basic definitions from probability theory which will be used in the thesis.

**Definition A.8** *Consider a sequence of $n$ independent trials. Each trial has two possible outcomes, a* **success** *and a* **failure**. *The probability that the outcome is a success is denoted by $p$ while that of a failure is denoted by $q \equiv 1 - p$. The probability that $k$ of these $n$ trials results in a success will be denoted by $b(k; n, p)$ and is given by:*

$$b(k; n, p) = \binom{n}{k} p^k (1-p)^{n-k}.$$

*The expected value of the random variable $k$ is given by $np$ and the variance is given by $npq$.*

The following definitions are from Complexity theory.

**Definition A.9** *Let $f(n)$, $g(n)$ be functic ıs from the positive integers to the positive reals.*

**(a)** *We write $f(n) \in O(g(n))$ if there exists a constant $c > 0$ such that, for large enough $n$, $f(n) \leq cg(n)$.*

**(b)** *We write* $f(n) \in \Omega(g(n))$ *if there exists a constant* $c > 0$ *such that, for large enough* $n$, $f(n) \geq cg(n)$.

**(a)** *We write* $f(n) \in \Theta(g(n))$ *if there exist constants* $c, c' > 0$ *such that, for large enough* $n$, $cg(n) \leq f(n) \leq c'g(n)$.

# Appendix B

# Proofs of Theorems

In this appendix we will present the MMR algorithm together with a proof of optimality. We will also include proofs of theorems.

## B.1 The MMR Algorithm

In this section we will introduce the Maximum Marginal Return (MMR) algorithm and prove its optimality.

Let $F : \Re^n \longrightarrow \Re$ be defined as:

$$F(\vec{x}) = \sum_{i=1}^{n} f_i(x_i) \qquad x_i \in Z_+, \quad i = 1, 2, \ldots, n$$

where each of the functions $f_i(x)$ has the property:

$$f_i(x-1) - f_i(x) \geq f_i(x) - f_i(x+1).$$

For any $m \in Z_+$ consider the following optimization problem:

$$\min_{\vec{x} \in Z_+^n} F(\vec{x}) \tag{B.1}$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_i = m.$$

The following algorithm, called the Maximum Marginal Return (MMR) algorithm, is optimal for this problem.

**procedure** MMR
**begin**
    $\vec{x} := [0, \ldots, 0]^T$

```
    for i := 1:n do Δ_i := f_i(x_i) − f_i(x_i + 1)
    for j := 1:m do
    begin
          Let k be such that Δ_k = max_i{Δ_i};
          x_k := x_k + 1;
          Δ_k := f_k(x_k) − f_k(x_k + 1);
    end
end
```

This is basically a greedy algorithm. In each iteration the index $k$ is found for which an increase in $x_k$ by unity produces the maximum decrease in the objective function. The value of $x_k$ is increased by one and the process is repeated until the constraint is satisfied. If the marginal return data is stored in heaps then the initial heap data structure requires $O(n)$ operations to construct. In each iteration the maximum marginal return item is removed and the heap must be reorganized. This requires $O(\log n)$ operations. Since there are $m$ iterations, the worst case complexity of the algorithm is $O(n + m \log n)$. We will next prove the optimality of the algorithm.

**Theorem B.1** *The solution produced by the MMR algorithm is optimal for problem B.1.*

*Proof:* The proof of this theorem for the special case in which the functions $f_i$ have the form $f(x) = V(1-p)^x$ is given in [4]. We have generalized their proof. The proof of the theorem will be by induction. Note that the theorem is trivially true for the case $m = 1$. Assume that it is true for $m = \bar{m}$. Denote the optimal solution for this case by $\vec{\bar{x}}$. Now suppose that:

$$f_k(\bar{x}_k) - f_k(\bar{x}_k + 1) = \max_i\{f_i(\bar{x}_i) - f_i(\bar{x}_i + 1)\} \tag{B.2}$$

Let us denote the solution produced by the algorithm for the case $m = \bar{m} + 1$ by $\vec{x}^*$. Note that $\vec{x}^* = \vec{\bar{x}} + e_k$. Let $\vec{z}$ be any feasible solution to the problem with $m = \bar{m} + 1$ other than $\vec{x}^*$. There must exist some $j$ such that $z_j > x_j^* \geq \bar{x}_j$. Let $\vec{\bar{\bar{x}}} = \vec{\bar{x}} + e_j$. We have

$$F(\vec{z}) = F(\vec{z} - e_j) - [f_j(z_j - 1) - f_j(z_j)]. \tag{B.3}$$

We also have that

$$F(\vec{\bar{\bar{x}}}) = F(\vec{\bar{x}}) - [f_j(\bar{x}_j) - f_j(\bar{x}_j + 1)]. \tag{B.4}$$

Since $\bar{\vec{x}}$ is optimal for the case $m = \bar{m}$ we have

$$F(\vec{z} - e_j) \geq F(\bar{\vec{x}}) \tag{B.5}$$

and by the assumptions on the functions $f_i$ stated in the problem we have

$$f_j(\bar{x}_j) - f_j(\bar{x}_j + 1) \geq f_j(z_j - 1) - f_j(z_j) \tag{B.6}$$

since $z_j > \bar{x}_j$. If we subtract B.4 from B.3 and use the inequalities B.5 and B.6 then we can show that $F(\vec{z}) \geq F(\bar{\vec{x}})$. Furthermore one can use B.2 to show that $F(\vec{x}^*) \leq F(\bar{\vec{x}})$. We therefore have that

$$F(\vec{z}) \geq F(\vec{x}^*).$$

This implies that the solution $\vec{z}$ is no better than the solution $\vec{x}^*$ obtained by the algorithm in the theorem. Since $\vec{z}$ can be any feasible solution we conclude that the solution obtained by the algorithm for the case $m = \bar{m} + 1$ is optimal. Therefore, by induction, the theorem is true for all $m \geq 0$. ■

## B.2 Proof of Lemma 3.2

We need to prove that the function, $q^{M-m_t} F_s^*(m_t)$, is convex with respect to $m_t$. The function $F_s^*(m_t)$ is the optimal cost of the static problem with two targets and $m_t$ weapons. Let us define

$$G(m) \equiv q^{M-m} F_s^*(m).$$

Since $F_s^*$ is defined only for integral values of $m$ we need to show that

$$G(m-1) - 2G(m) + G(m+1) \geq 0$$

For any fixed value of $m > 0$ we can write:

$$F_s^*(m-1) = V_1 q^{x_1} + V_2 q^{x_2}.$$

If the number of weapons is increased by one then the additional weapons will be added to the target with the maximmum marginal return. Let us suppose, for convenience, that

$$V_1 p q^{x_1} \geq V_2 p q^{x_2} \tag{B.7}$$

so that the additional weapon goes to target 1 then

$$F_s^*(m) = V_1 q^{x_1+1} + V_2 q^{x_2}.$$

If the number of weapons is further increased by 1 then we need to consider 2 cases.

CASE I: $V_2 p q^{x_2} \geq V_1 p q^{x_1+1}$.

In this case target 2 has the maximum marginal return so that

$$F_s^*(m+1) = V_1 q^{x_1+1} + V_2 q^{x_2+1}.$$

We can now write

$$\begin{aligned} G(m-1) - G(m) + G(m+1) &= q^{M-m-1}[q^2 F_s^*(m-1) - 2q F_s^*(m) + F_s^*(m+1)] \\ &= q^{M-m-1}[V_1 q^{x_1+1}(1-q) + V_2 q^{x_2+1}(q-1)] \\ &= q^{M-m}[V_1 p q^{x_1} - V_2 p q^{x_2}] \\ &\geq 0. \end{aligned}$$

The last inequality is a result of the inequality in B.7.

CASE II: $V_2 p q^{x_2} \leq V_1 p q^{x_1+1}$.

In this case target 1 has the maximum marginal return so that

$$F_s^*(m+1) = V_1 q^{x_1+2} + V_2 q^{x_2}.$$

We can now write

$$\begin{aligned} G(m-1) - G(m) + G(m+1) &= q^{M-m-1}[q^2 F_s^*(m-1) - 2q F_s^*(m) + F_s^*(m+1)] \\ &= q^{M-m-1}[V_2 q^{x_2}(q^2 - 2q + 1)] \\ &= q^{M-m-1} V_2 q^{x_2} p^2 \\ &\geq 0. \end{aligned}$$

Therefore the function $G(m)$ is convex which completes the proof.

# Bibliography


[1] M. Athans, "Command and control (C2) theory: a challenge to control science," *IEEE Transactions on Automatic Control*, vol. AC-32, pp. 286–293, Apr. 1987.

[2] S. F. Lloyd and H. S. Witsenhausen, "Weapons allocation is NP–Complete," in *Proceedings of the 1986 Summer Conference on Simulation*, (Reno, Nevada), Jul. 1986.

[3] H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*. New Jersey: Prentice Hall, Inc., 1981.

[4] G. G. denBroeder, R. E. Ellison, and L. Emerling, "On optimum target assignments," *Operations Research*, vol. 7, pp. 322–326, 1959.

[5] J. D. Kattar, "A solution of the multi–weapon, multi–target assignment problem," Working Paper 26957, MITRE, Feb. 1986.

[6] S. M. Matlin, "A review of the literature on the missile-allocation problem," *Operations Research*, vol. 18, pp. 334–373, 1970.

[7] A. R. Eckler and S. A. Burr, *Mathematical Models of Target Coverage and Missile Allocation*. Alexandria, Va.: Military Operations Research Society, 1972.

[8] R. M. Soland, "Optimal terminal defense tactics when several sequential engagements are possible," *Operations Research*, vol. 35, pp. 537–542, Aug. 1987.

[9] S. A. Burr, J. E. Falk, and A. F. Karr, "Integer prim-read solutions to a class of target defense problems," *Operations Research*, vol. 33, pp. 726–745, 1985.

[10] D. Castañon. Personal Communications.

[11] R. T. Rockafellar, *Network Flows and Monotropic Optimization. Wiley-Interscience*, New York: Wiley, 1984.

[12] D. P. Bertsekas, "Notes on nonlinear programming and discrete-time optimal control," LIDS Report LIDS-R-919, Massachusetts Institute of Technology, Cambridge, MA, Jan. 1983.

[13] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. New Jersey: Prentice Hall, 1982.

[14] D. Orlin, "Optimal weapons allocation against layered defenses," *Naval Research Logistics*, vol. 34, pp. 605–617, 1987.

[15] G. E. Pugh, "Lagrange multipliers and the optimal allocation of defense resources," *Operations Research*, vol. 12, pp. 543–567, 1964.

[16] D. P. Bertsekas, *Dynamic Programming*. New Jersey: Prentice Hall, Inc., 1987.

[17] J. Bracken, P. S. Brooks, and J. E. Falk, "Robust preallocated preferential defense," *Naval Research Logistics*, vol. 34, pp. 1–22, 1987.

[18] P. Hosein, M. Athans, and J. Walton, "Dynamic weapon-target assignment problems with vulnerable $C^2$ nodes," in *Proceedings of the 1988 Command and Control Symposium*, (Monterey, CA), June 1988.

[19] J. T. Walton and M. Athans, "Strategies for asset defense with precursor attacks on the command and control system," in *Proceedings of the 1989 Command and Control Symposium*, (Washington, DC), June 1989.

[20] W. Feller, *An Introduction to Probability Theory and Its Applications*. Vol. 2 of *Probability and Mathematical Statistics*, New York: Wiley, 3 ed., 1968.